DESIGN OF A COMPUTER AIDED OPTIMAL SPEED REGULATING SYSTEM FOR A BAR MILL MAIN DRIVE

A Thesis Submitted in Partial Fulfilment

for the Degree of

DOCTOR OF PHILOSOPHY

by BHATT PRAMOD CHANDRA PURUSHOTTAM

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR (U.P.)

December 1969

..... to VARSHA and RITU

CERTIFICATE

Certified that this work on "Design of a Computer Aided Optimal Speed Regulating System for a Bar Mill Main Drive" by Mr. P.C.P.Bhatt has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dec. 1, 1969

Dr. V. Rajaraman
Professor in Electrical Engineering,
Head, Computer Centre
Indian Institute of Technology
KANPUR (U.P.)

ACKNOWLEDGEMENTS

In the preparation of this thesis the following acknowledgements are due:

For guidance and over-all supervision	to	Prof. V. Rajaraman
For encouragement during the progress of work	to	Prof. M.A. Pai
For discussions on theorems in Chapter III	to	Dr. S.K. Gupta
For discussions on trajectory-manipulation-technique	to	Dr. S.D. Agashe Dr. P.C. Das
For computation	to	Mr. R.N. Basu Mr. S. Kapoor Mr. S.G. Ravikumar Mr. V.S. Manian Dr. Muthukrishnan
For help in preparation of the manuscript	to	Dr. V. Rajaraman Dr. V.P. Sinha Mr. P.R.K. Rao
For typing	to	Mr. K.N. Tewari
For the help in removing errors from the type-script	to	Mr. R.K.P. Bhatt Mr. Goel, R.C. Mr. Apte, P.R.

TABLE OF CONTENTS

LIST OF SYMBOLS AND CHARACTERS

Symbol	Word Description
a	a scalar
a_{ij}	machine parameters
A	weighting factor in performance index
	subset of set X
b_i	coefficient of forcing function in state equations
B	weighting factor in performance index
	subset of set Y
C	convex subset of X
D	weighting factor in performance index
D_G	Gateaux differential
E	performance index
$\overset{o}{E}$, $\min E$	optimum E
$f(\ )$	function of ( )
f	as a subscript for variables associated with field
F	integrand in performance index
g	as a subscript for a guess
h	a member of solution set S
	increment to evaluate Gateaux differential
h_n	n-th interval in a time slice
i	as a super-script denotes iteration number
	as a subscript denotes i-th set in set X

I_a	armature current
I_{fg}	generator field current
I_{fm}	motor field current
I_{ag}	guess of armature current trajectory
I_{fmg}	guess of motor field current trajectory
j	as a subscript denotes the j-th component of i-th set
J	index in a DO LOOP
	motor rotor and mill inertia
k	as a super-script denotes iteration number
K_i, K_{ij}	auxiliary time variable function
KI,KQ	intermediate coefficients in Gill's routine
L_a	armature loop inductance
L_{ag}	generator armature inductance
L_{am}	motor armature inductance
n	as a subscript denotes interval number
N_g	speed of rotation of the generator
N_m	speed of rotation of the motor
N_{md}	desired speed of rotation
N_{mg}	approximation to motor speed N_m
q_{ij}	locally defined variables
p	adjoint vector
	perturbed trajectory
r_a	armature loop resistance
r_{ag}	generator armature resistance
r_{am}	motor armature resistance

S	solution set
t	time
T_L	load torque
T_p, T_s	through primary and secondary variables
u	unit matrix
	control vector
V	convex real-valued functional
	voltage
V_f	field voltage
V_{fg}	generator field voltage
V_{fm}	motor field voltage
V_t	terminal voltage
V_{td}	desired terminal voltage
w_i	weighting factor less than 1
x	state vector
x_i	i-th component of state vector
x_{ig}	guess of x_i
X	domain set
y	augmented state and adjoint vector
Y	range set
α	a scalar
$\in$	belongs to the set
	small constant
	neighbourhood

μ	a scalar
σ	running variable
η	a perturbation
$\phi(\;\;)$	function of ( )
	generally denotes flux
∇	differential operator
∇_G	Gateaux differential operator
a.b	scalar multiplication of vectors a,b
a * b	a multiplied by b, also used in the sense of a .b
a / b	a divided by b
d x/dt, $\dot{x}$	derivative of x with respect to t.
$\forall$	for all.
$\subseteq$	subset of the set

SYNOPSIS

Bhatt Pramod Chandra Purushottam
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

December 1969

DESIGN OF A COMPUTER AIDED OPTIMAL SPEED REGULATING SYSTEM FOR A BAR MILL MAIN DRIVE

In this thesis an optimal design of a speed regulating system for a bar mill main drive has been obtained.

In earlier designs of speed regulating systems, although concepts of mathematical modelling were used, the merit of the control configuration was judged by experimental observations and analog computer simulation. This left open the question of optimality.

In this thesis, a new technique of designing speed regulating systems is evolved by using dynamic programming concepts. Use of dynamic programming enables one to answer questions about the goodness of design with certainty.

There are two notable features of modelling of a main drive. One is that, for both armature current and air gap flux varying with time, a product term appears in the state equations. Another is that, the field system in main drive has a nonlinear B-H characteristic.

It would appear that no explicit optimisation technique has so far been suggested for systems described by state equations that involve product terms. However, if one of the two variables (armature current and air gap flux) is time invariant, then a single valued piecewise linear approximation of B-H curve may be used. Consequently, optimisation techniques for "nearly linear" systems may be employed in this case also.

The optimisation method developed here is based upon the premise that the product term can be approximated by a weighted linear combination of both the terms with time varying coefficients. Starting with an initial guess on the trajectories of both the variables and using them as time varying coefficients, the corresponding set of time varying linear or non-linear differential equations is constructed. The approximation of product terms is improved iteratively by taking Gateaux gradient functional of a non-negative function along each of the guessed trajectories. The optimal solution is obtained by using parametric expansion in each iteration. Thus one essentially solves a sequence of optimal problems which converge to the required solution.

A control program implementing this optimisation process has been written in a form that is adaptable for use in control computers. On-line execution of this program for both generator field control mode as well as motor field control mode of operation has also been discussed.

CHAPTER - I

INTRODUCTION

Following the publication of works of Pontryagin[1] and Bellman[2], the term "optimum-design" acquired definite meaning in theoretical design of control systems. The engineering translations of these designs seem nearly feasible*, only when a fast interacting computer system is used as a control tool.

This thesis concerns with design of a computer-aided-optimal speed regulating system of a certain bar mill drive. The optimality of operation is ensured by using Bellman's dynamic programming concepts in evolving the control logic. It is assumed that for on-line operations a process control computer would be available.

In what follows an over-view of the thesis is provided.

1.1 Logic for System Design[1,2,3]

The logic often used in attempting optimum system design is:
if feasible solutions exist (satisfying design specifications) then there exists an optimum solution which not only satisfies the design specifications but is also optimum with respect to some criterion. This argument provides the basis for the present investigation.

---

* nearly ... because of finite measurement delays and processing times involved.

1.2 Feasible Solutions

Both a.c. and d.c. drives have been employed in steel rolling mills. However, experience[4,12] has proven that the d.c. drive systems are to be preferred. Among the d.c. drive systems there are two arrangements: one has a three machine complex while the other one has one machine and heavy rectifying equipment accompanying it. The present investigation concerns a three machine configuration as depicted in Fig.1-1.

The two modes of operation[4,5,6,7] for the arrangement in Fig.1-1 are

1) Below the base speed of operation the error signal (difference between desired and actual speeds) is fed to the channel corresponding to generator field control,

2) Above the base speed the error signal is fed to the channel corresponding to motor field control.

The magnitude of the terminal voltage is utilised to exercise switching of error signal between the two modes of control. All the schemes, cited above, confirm to this arrangement.

In a recent study[7] (Appendix A) on a control scheme similar to Plaskett's[6], the following was observed:

a) On employing field forcing units which are capable of supplying high d.c. voltage for short periods, one can obtain fast field current response. This can be explained by observing the solution of a first order differential equation as detailed in Fig.1-2.

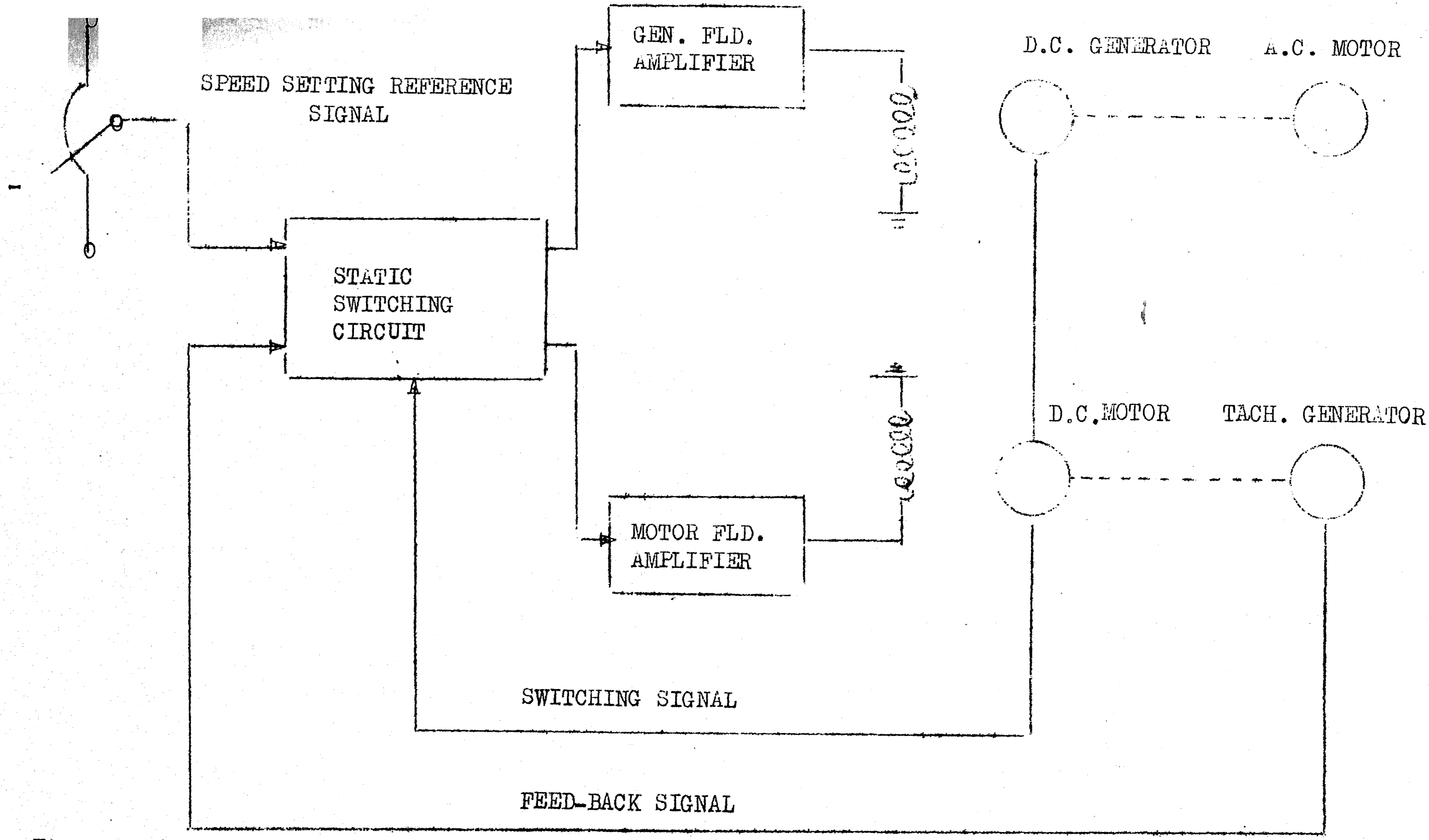
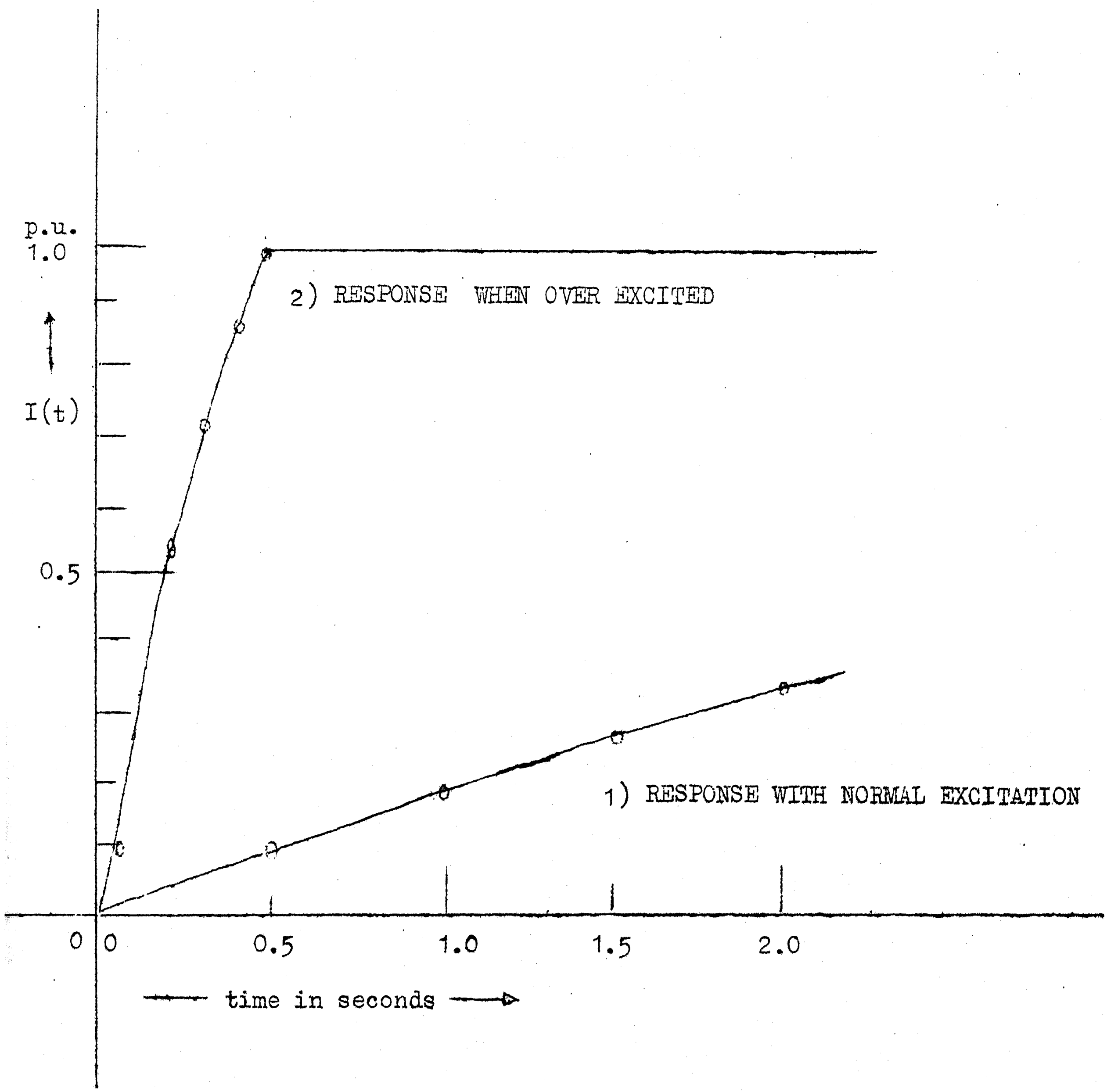


Figure 1-1

The basic control configuration of three machine complex.



1) $V(t) = U(t)$

2) $V(t) = 10 * U(t) \quad t < 0.5$, $V(t) = U(t) \quad$ for $t \geq 0.5$

The model of field coil :-

$$V(t) \rightarrow \boxed{\frac{1}{0.2s + 1}} \rightarrow I(t)$$

Figure 1-2

Current response of a field coil with and without normal excitation.

b) It is easy to achieve the separation of two modes of control by appropriate hardware.

The studies cited above demonstrate that feasible solutions exist but the question whether these are also optimal, in some sense, remains unanswered.

1.3 Investigation for Optimum

Having confirmed (Appendix A) that for a particular bar mill main drive feasible solutions existed, it was decided to investigate if it was possible to design a computer aided optimal speed control system. The use of computer as a control aid is suggested because a computer system would be available, as such, for economic reasons[9,10].

The Modelling of Mill Main Drive

The mill drive has an overall dynamics of fourth order, but because in different time slices, either generator field current or motor field current is constant, one essentially has a third order plant. The state equations of the process, when generator field control is exercised, are linear if one agrees to linearise the generator field B-H characteristics (a reasonable assumption). The state equations of the process, when motor field control is exercised, are non-linear and involve both a product type of non-linearity as well as the functional non-linearity of the B-H curve. As the field weakening/strengthening (as the need be) is exercised, over a large range (usual range being 0.5 p.u. to 1.5 p.u.), the functional non-linearity of motor field systems cannot be over simplified.

Performance Index

For either mode of operation, the criterion of performance was chosen of the form 1.3.1.

$$P.I. = \int_0^T \left[A * (N_m - N_{md})^2 + B * v_f + v_f^2 + D * (V_t - V_{td})^2 \right] dt \qquad 1.3.1$$

Flexibility in selection of the coefficients A, B and D is to be observed[13,14,15,16] to finally yield an acceptable design.

Optimisation Procedure

For the generator field control (linear control problem), Merriam's[10] parametric expansion technique has been used.

To compute optimal control law for the motor field control mode of operation a new technique has been suggested in this thesis. The method requires replacement of product terms in the state equations by a weighted linear combination of both the terms as explained in equation 1.3.2.

$$\left.\begin{array}{l} x_1(t) * x_2(t) = w * x_{1g}(t) * x_2(t) + (1 - w) * x_1(t) * x_{2g}(t) \\ \text{where } 0 < w < 1 \text{ and} \\ \qquad x_{1g}, x_{2g} \text{ refer to guesses to be improved iteratively} \end{array}\right\} \qquad 1.3.2$$

Now the optimal solution of the system equations with trajectory approximation as in equation 1.3.2, are obtained using parametric expansion method. Based on the solution the approximation in 1.3.2 is modified. The trajectories corresponding to $(x_{1g}, x_{2g}, \ldots, x_{ng})$ are manipulated by using Gateaux derivative of a positive-semidefinite

functional for each t in the interval $[0,T]$. The Gateaux differential of this functional equals zero over the entire time interval when the approximation is exact. One major advantage of this approximation is that Merriam's parameteric expansion can be used to obtain two initial value problems from the corresponding Hamilton Jacobi equations and solution of a two point boundary value problem is avoided.

1.4 Main Contributions of the Thesis

The object set for the thesis was to evolve an optimal design for a bar mill main drive speed control system and this has been achieved. The optimal solution is obtained for continuous generator/motor field voltages, as it was assumed that continuous voltages would be available as forcing functions.

In the course of this investigation a powerful trajectory manipulation technique, which can be easily implemented, using a digital computer, has been developed. The technique may be employed to optimise control problems characterised by multiplicative and functional (single-valued) non-linearities in system equations. Adequate proofs for the algorithm on optimising procedure are also provided.

1.5 Outline of other Chapters

In Chapter II, the mathematical model associated with the bar mill main drive has been derived. The performance requirements on speed control system are spelled out next and a complete problem statement follows immediately.

To frame an appropriate context for the method suggested for the problems stated in Chapter II, a brief review of many algorithms available in the literature is presented in Chapter III. In the last two sections of the chapter, the new trajectory manipulation technique is discussed with complete details. Definitions and theorems concerning the proposed technique are also given.

Chapter IV primarily concerns with the computational aspects of the optimisation technique suggested in Chapter III for the problem stated in Chapter II. All the flow diagrams corresponding to computer program are described in detail. The computer results detailing the performance of the mill main drive are also given. The chapter ends with a discussion on on-line execution in a hypothetical situation, where a process control computer is available as a control tool.

In the last chapter the main conclusions from this investigation are highlighted and directions for further research are suggested.

The Appendix A, B and C contain material which is peripheral to main stream of thought in the thesis.

CHAPTER - II

THE PROBLEM STATEMENT

The purpose of this chapter is to obtain a mathematical model of the bar mill main drive and make a statement about the desired performance.

The modelling procedure employs an oriented system graph for the main drive. The cutset and circuit equations for the graph are substituted in the terminal equations to finally yield the state equations.

The design requirements are based on an industrial project report[30]. These specifications essentially define an admissible design.

To achieve an optimum admissible design, a criterion of performance is chosen and a logic of iteration is followed to arrive at appropriate weighting factors to be associated with each of the constituents of the performance criterion. In the same section the protection scheme for the main drive is also discussed because the logic of iteration (at computational stage) is ~~based on~~ related with the protective features (at operational stage) to be finally associated with the mill main drive.

2.1 Derivation of State Model

The state model for the system can be obtained by following graph-theoretic procedure suggested by P.H.O'N.Roe[32]. The method involves systematic substitution of circuit and cutset equations in the terminal equations to yield state equations for the given system. The forest is chosen so that most voltage drivers and capacitors are

included in the forest while most of the current drivers and inductors are consigned to co-forest. For non-degenerate circuits, capacitor voltages and inductor currents finally turn out to be state variables and, therefore, should be preferably measurable. In what follows the chosen forest is referred to as the formulation forest, the tree branch voltages and chord currents are called primary variables and the rest are called secondary variables.

The system graph corresponding to a mill drive is shown in Fig.2-1. The formulation forest is detailed by the solid lines in Fig.2-1. The circuit and cutset equations can be written by arranging the primary and secondary variables to yield equations 2.1.2 having the form 2.1.1.

$$\left.\begin{array}{l}
\text{Circuit Equations} \\
\begin{bmatrix} B & \vdots & U \end{bmatrix} \begin{bmatrix} A_p \\ \hdashline A_s \end{bmatrix} = 0 \\
\text{Cutset Equations} \\
\begin{bmatrix} U & \vdots & A \end{bmatrix} \begin{bmatrix} T_p \\ \hdashline T_s \end{bmatrix} = 0
\end{array}\right\} \quad .. \quad .. \quad .. \quad 2.1.1$$

where A and T refer to across and through variables and

subscripts p , s are used to distinguish primary/secondary variables.

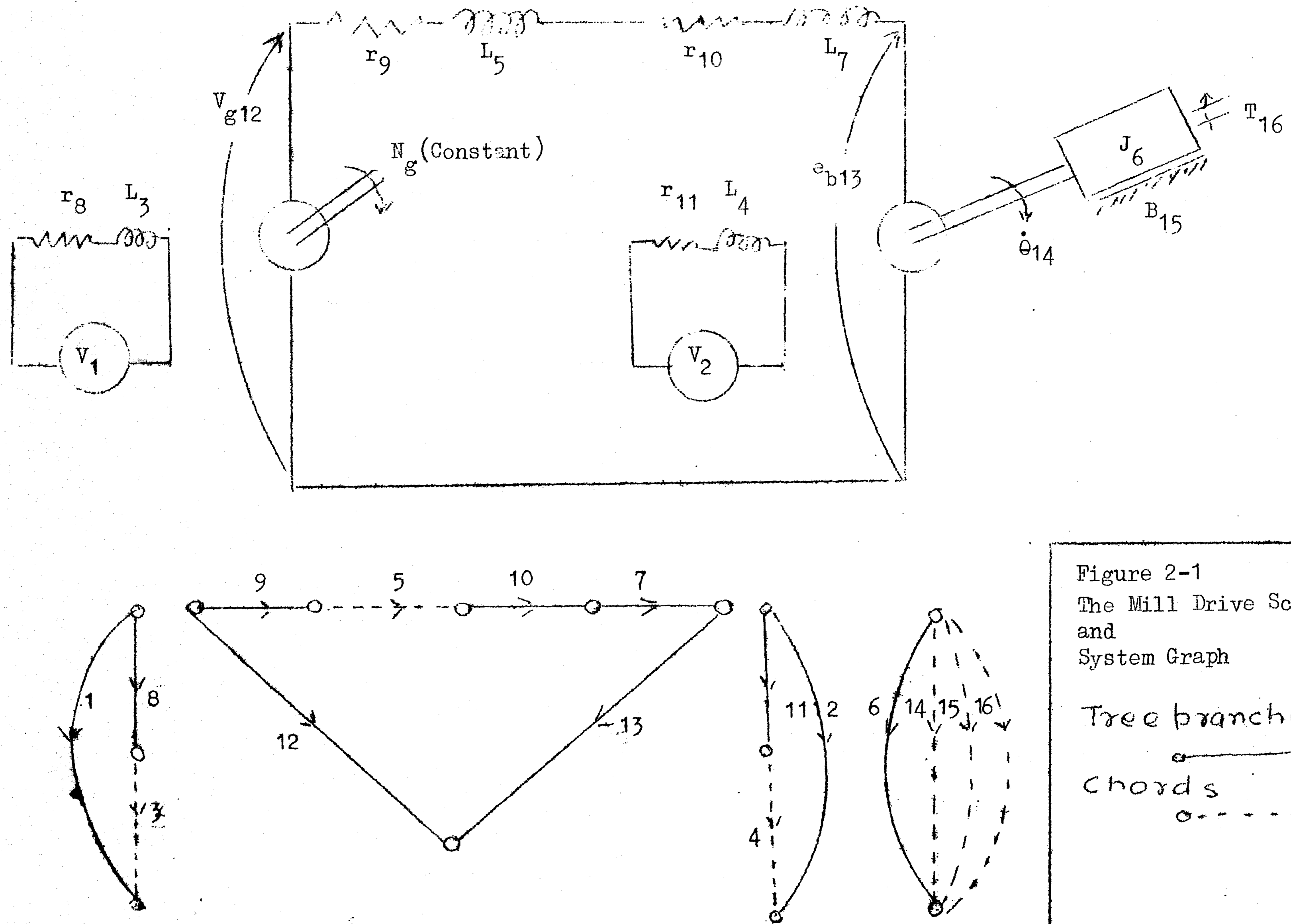


Figure 2-1
The Mill Drive Schematic
and
System Graph

The circuit equations are

1	2	6	7	8	9	10	11	12	13	3	4	5	14	15	16
-1				1						1					
	-1						1				1				
			1		1	1		-1	1			1			
		-1											1		
		-1												1	
		-1													1

$$\begin{bmatrix} V_1 \\ V_2 \\ \dot{\theta}_6 \\ V_7 \\ V_8 \\ V_9 \\ V_{10} \\ V_{11} \\ V_{12} \\ V_{13} \\ V_3 \\ V_4 \\ V_5 \\ \dot{\theta}_{14} \\ \dot{\theta}_{15} \\ \dot{\theta}_{16} \end{bmatrix} = 0$$

The cutset equations are

1	2	6	7	8	9	10	11	12	13	3	4	5	14	15	16
1										1					
	1										1				
		1											1	1	1
			1									-1			
				1						-1					
					1							-1			
						1						-1			
							1				-1				
								1				1			
									1			-1			

$$\begin{bmatrix} I_1 \\ I_2 \\ T_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \\ I_{11} \\ I_{12} \\ I_{13} \\ I_3 \\ I_4 \\ I_5 \\ T_{14} \\ T_{15} \\ T_{16} \end{bmatrix} = 0 \qquad \text{.. 2.1.2}$$

Now on writing the terminal equations explicitely in primary variables we have

$$
\left.
\begin{array}{l}
\begin{bmatrix} d\,I_3/dt \\ d\,I_4/dt \\ d\,I_5/dt \\ d\,\dot{\theta}_6/dt \end{bmatrix}
=
\begin{bmatrix} 1/L_3 & & & \\ & 1/L_4 & & \\ & & 1/L_5 & \\ & & & 1/J_6 \end{bmatrix}
\begin{bmatrix} V_3 \\ V_4 \\ V_5 \\ T_6 \end{bmatrix}
\\[2ex]
V_7 = L_7 * (d\,I_7/dt)
\\[2ex]
\begin{bmatrix} V_8 \\ V_9 \\ V_{10} \\ V_{11} \\ T_{15} \end{bmatrix}
=
\begin{bmatrix} r_8 & & & & \\ & r_9 & & & \\ & & r_{10} & & \\ & & & r_{11} & \\ & & & & B_{15} \end{bmatrix}
\begin{bmatrix} I_8 \\ I_9 \\ I_{10} \\ I_{11} \\ \dot{\theta}_{15} \end{bmatrix}
\\[2ex]
\begin{bmatrix} V_{12} \\ V_{13} \\ T_{14} \end{bmatrix}
=
\begin{bmatrix} K_{12} * \phi(I_3) \\ K_{13} * \dot{\theta}_{14} * \phi(I_4) \\ -K_{13} * I_{13} * \phi(I_4) \end{bmatrix}
\end{array}
\right\}
\quad .. \qquad 2.1.3
$$

Using circuit and cutset equations we can eliminate secondary variables from the terminal equations 2.1.3 so that

$$
\begin{bmatrix} V_8 \\ V_9 \\ V_{10} \\ V_{11} \\ T_{15} \end{bmatrix}
=
\begin{bmatrix} r_8 & & & \\ & & r_9 & \\ & & r_{10} & \\ & r_{11} & & \\ & & & B_{15} \end{bmatrix}
\begin{bmatrix} I_3 \\ I_4 \\ I_5 \\ \dot{\theta}_6 \end{bmatrix}
\quad .. \quad .. \qquad 2.1.4
$$

$$\begin{bmatrix} V_{12} \\ V_{13} \\ T_{14} \end{bmatrix} = \begin{bmatrix} K_{12} * \phi(I_3) \\ K_{13} * \phi(I_4) * \dot{\theta}_6 \\ -K_{13} * \phi(I_4) * I_5 \end{bmatrix}$$

$$\begin{bmatrix} d\, I_3/dt \\ d\, I_4/dt \\ d\, I_5/dt \\ d\, \dot{\theta}_6/dt \end{bmatrix} = \begin{bmatrix} 1/L_3 & & & \\ & 1/L_4 & & \\ & & 1/L_5 & \\ & & & 1/J_6 \end{bmatrix} \begin{bmatrix} V_1 - V_8 \\ V_2 - V_{11} \\ V_{12} - V_7 - V_9 - V_{10} - V_{13} \\ -(T_{14} + T_{15} + T_{16}) \end{bmatrix} \qquad \ldots \ldots \quad 2.1.4$$

From the equations 2.1.4, the state equations (equation 2.1.5) can be derived

$$\left.\begin{aligned}
d\, I_3/dt &= - r_8 * I_3 / L_3 + V_1 / L_3 \\
d\, I_4/dt &= - r_{11} * I_4 / L_4 + V_2 / L_4 \\
d\, I_5/dt &= -(r_9 + r_{10}) * I_5 / (L_5 + L_7) - K_{13} * \dot{\theta}_6 * \phi(i_4) / (L_5 + L_7) \\
&\quad + K_{12} * \phi(I_3) / (L_5 + L_7) \\
d\, \dot{\theta}_6/dt &= K_{13} * \phi(I_4) * I_5 / J_6 - B_{15} * \dot{\theta}_6 / J_6 - T_{16} / J_6
\end{aligned}\right\} \quad 2.1.5$$

2.2 The Normalised State Equations

The following system data is assumed to be given :

Gen. ratings : 2040-kw, 400-volts, 750 rpm

$r_{ag} = r_9 = 0.00165$ ohms, $L_{ag} = L_5 = 0.1075$ mh

$r_{fg} = r_8 = 1.28$ ohms, $L_{fg} = L_3 = 4.4$ h

rated field current 50 amps.

Motor ratings : 1875 kw, 400 volts, 80/160 rpm

$r_{am} = r_{10} = 0.00299$ ohms, $L_{am} = L_7 = 0.313$ mh

$r_{fm} = r_{11} = 1.0$ ohm, $L_{fm} = L_4 = 5.03$ h

rated field current 100 amps

rotor inertia 10,000 kg.m^2

mill inertia 2820 kg.m^2

Total mill inertia = J_6 = 12820 kg.m^2

From the given system data normalised state equations can be obtained by defining following per unit* quantities.

For the generator field circuit :

Per unit voltage = 47.5 volts, Per unit current = 50.0 amps.

Time constant = (1/0.288) seconds

For the armature circuit :

Per unit voltage = 386 volts, Per unit current = 5140 amps.

Time constant = (1/11.09) seconds

For motor field circuit :

Per unit voltage = 100 V, Per unit current = 100 amps.

Time constant = (1/0.218) seconds

For the mechanical circuit :

Per unit speed = 80 rpm

Per unit torque = p.u. power/(p.u. rps x 2 π)

= 236096 newton-meters

Per unit inertia = p.u. torque/p.u. acceleration

= 28095.42 kg.m^2

Total mill inertia is then 0.446 p.u.

---

* Elsewhere in the thesis per unit quantities are also referred to as base quantities.

And the normalised state equations can be written as

$$\left.\begin{aligned}
d\, I_{fg}/dt &= -\,0.288 * I_{fg} + 0.288 * V_{fg} \\
d\, I_{fm}/dt &= -\,0.218 * I_{fm} * K_{fm} + 0.218 * V_{fm} * K_{fm} \\
d\, I_a/dt &= -\,11.09 * I_a + 11.09 * I_{fg} - 11.09 * \phi(I_{fm}) * N_m \\
d\, N_m/dt &= 2.24 * I_a * \phi(I_{fm}) - 2.24 * T_L \\
K_{fm} &= I_{fm} / \phi(I_{fm})
\end{aligned}\right\} \quad .. \; 2.2.1$$

An assumption of single valued relationship between (ϕ_{fg}, I_{fg}) and (ϕ_{fm}, I_{fm}) leads to the choice of I_{fg} and I_{fm} as measurable state variables besides simplifying the overall structure of state equations*. In equation 2.2.1 the (ϕ_{fg}, I_{fg}) characteristic is assumed to be linear while the (ϕ_{fm}, I_{fm}) characteristic is assumed to be piece-wise linear (K_{fm} adjusts fld.time const. in each segment).

For either generator field control mode of operation or motor field control mode of operation the state equations can be obtained from equations 2.2.1 by observing that

1) For generator field control mode

$$d\, I_{fm}/dt = 0 \quad .. \quad .. \quad .. \quad .. \quad 2.2.2$$

2) For motor field control mode

$$d\, I_{fg}/dt = 0 \quad .. \quad .. \quad .. \quad .. \quad 2.2.3$$

---

* The double valued approximation of (ϕ, I) characteristics require that flux be considered as a function of both I and (d I/dt). A hysteresis curve can be simulated by integrating an equation of the following form :

$$d\,\phi/dI = 1 / (1 - a * I^2 + b * I^4) + c * d\, I/dt,$$

where a, b and c are constants. Constants a, b and c can be obtained from $\phi(I, t)$ at t = 0, I(0) and d I(0+)/dt. A sample example with a = 0.25, b = 4 and c = 0.06 and $\phi(0, 0) = 0.2105$ yields a symmetrical hysteresis curve.

2.3 The Statement of the Problem

The problem can be stated now as : Given the state model for the bar mill drive and specified rolling torque (Appendix B) obtain a control configuration to achieve[30]

a) an acceleration rate of 0.5 p.u. to race to base speed from standstill with load (ref. Fig.2-2),

b) an acceleration rate of 0.5 p.u. to race to top speed from base speed with load (ref. Fig.2-2),

c) a recovery time of not greater than 0.5 seconds for impact loading or load throw-off at top speed,

d) speed regulation 4%

ensuring the following,

1) the armature current does not exceed 1.9 p.u. and persist beyond 0.5 seconds duration,

2) the maximum rate of change of armature current does not exceed 2.0 p.u./sec. and persist beyond 0.5 seconds duration at that rate,

3) the regenerative current does not exceed 0.9 p.u. and persist beyond 0.5 seconds during the field weakening period,

4) the rate of change of either of the field currents does not exceed 8.0 p.u./sec. and persist at that rate beyond 0.2 seconds.

The restrictions (1 - 4) are dictated by machine design and often stabilising components are used[7] in addition to the regular protection equipment to ensure safe operation of the mill. It is assumed that continuous forcing functions will be available.

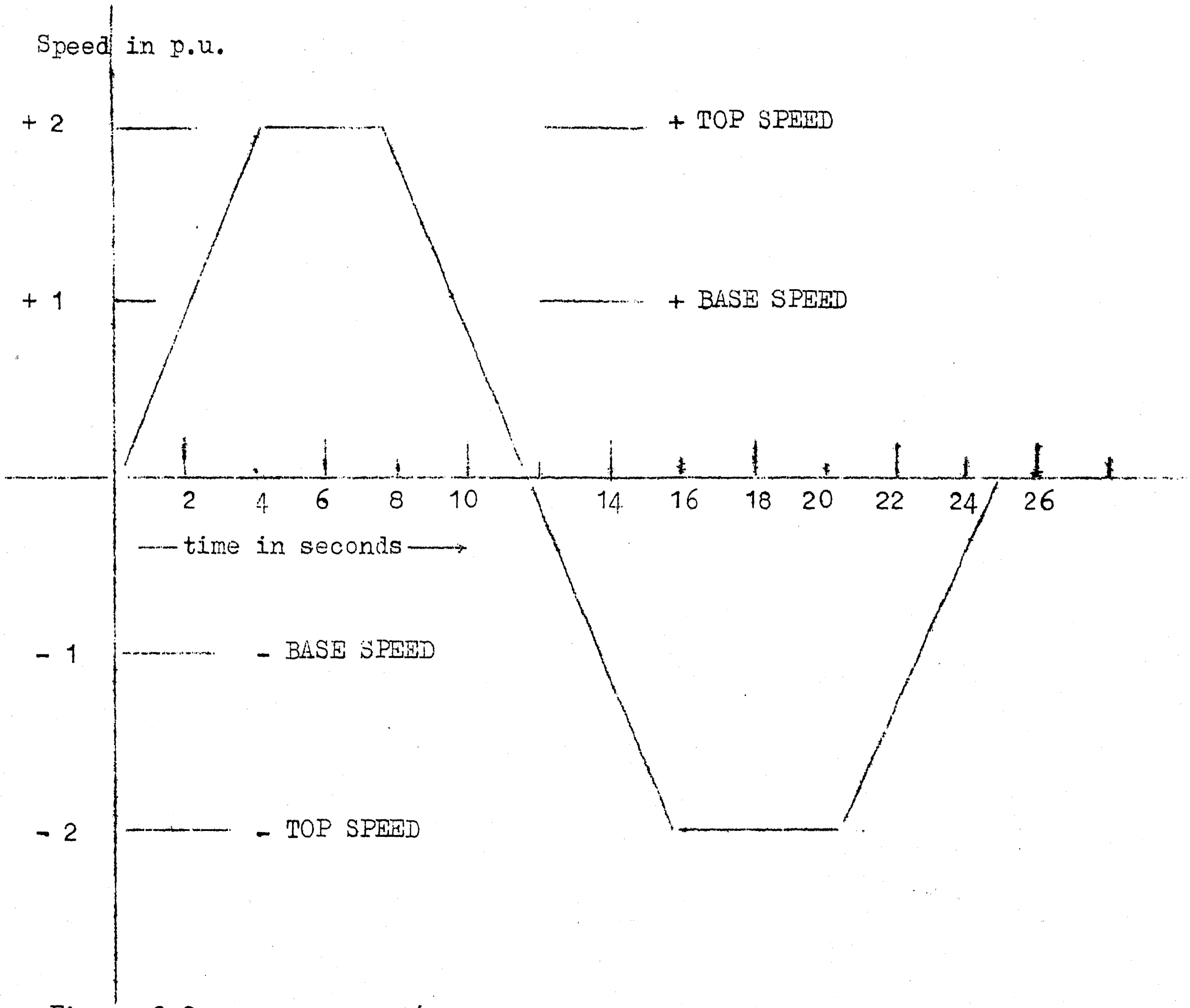


Figure 2-2

Speed versus time characteristics.

2.4 The Choice of a Criterion of Optimality and the Logic of Iterations on Weighting Factors

In arriving at the form of criterion of optimality some flexibility has to be observed to ensure physical realisability[13,14,15,16].

An optimality criterion* of the form (2.4.1) was chosen,

$$E = \min \int_0^T \left[A * (N_m - N_{md})^2 + B * V_f + V_f^2 + D * (V_t - V_{td})^2 \right] dt \quad (2.4.1)$$

This choice was based on the following considerations :

i) The primary variable of interest is speed and a measure of deviation of actual speed from desired speed should be included,

ii) For hard-ware realisability there shall be a specified limit on the value of forcing function so a function which reflects steep penalty as the magnitude of V_f increased should be included,

iii) The inter-coil voltage should not exceed a specified limit for safe operation and this is reflected in the term containing V_t,

iv) The term $B * V_f$ has been included to slightly under weight negative values of V_f so as to have the peak - 10 p.u. reached (without raising the value of A very high).

As such, there is no general procedure[15,16] to decide on numerical values of A, B and D. However a convenient rule would be to follow the algorithm described by the decision table[17] (Table 2-1). The condition entries are particular clock readings and maximum values of forcing voltage. The clocks are simulated in the program to check on operational constraints described in Section 2.3. In practice these clocks would be expected to provide protective features only.

---

* It might seem that the functional E does not necessarily remain non-negative and convex. As during the time V_f is negative first term dominates in the integrand, the function E remains non-negative a cont. to II

Table 2-1

Key w = maximum permissible reading any clock

x = reading on any of the clocks any time

y = maximum magnitude of forcing function

z = maximum permissible magnitude of forcing function

1 = the particular condition exists

Condition Stub	Condition entries			
$x < w$	1	1		
$x > w$			1	1
$y < z$	1		1	
$y > z$		1		1
Action on A	increase	decrease	decrease	decrease
Action on B or Action on D	decrease	increase	increase	increase

The increments can be based on an expression of the form 2.4.2 to finally yield a value for "A" so that $V_{f_{max}}$ has a peak value of 10.

$$A_{new} = A_{old} + (10. - V_{f_{max-old}}) * K(V_{f_{max}}) \quad .. \quad .. \qquad 2.4.2$$

where K is a multiplier depending upon the value of $V_{f_{max}}$.

From II-9

plot of E versus time would reveal its convex nature.

The clocks (Fig.2-3) record the time durations for which the derivatives of field current, armature current and speed of motor exceeds the specified limit. The clocks are reset, to read zero, only when 0.5 seconds have elapsed after it was last incremented. Thus ensuring that in no one second duration abnormal conditions existed beyond 0.5 seconds. The numerical values of 'A', 'B' and 'D' are so chosen that mill motor will not trip under normal operating conditions.

The finally accepted values of A, B and D which ensured normal operations are given in Table 2-2.

Table 2-2

Finally accepted values of A, B and D.

	Value of A	Value of B	Value of D
Gen. field control			
a) zero to base	4800	1	0
b) at top speed*	4800	0.25	200
Motor field control	8800	5	100

* For both impact load on and load throwoff.

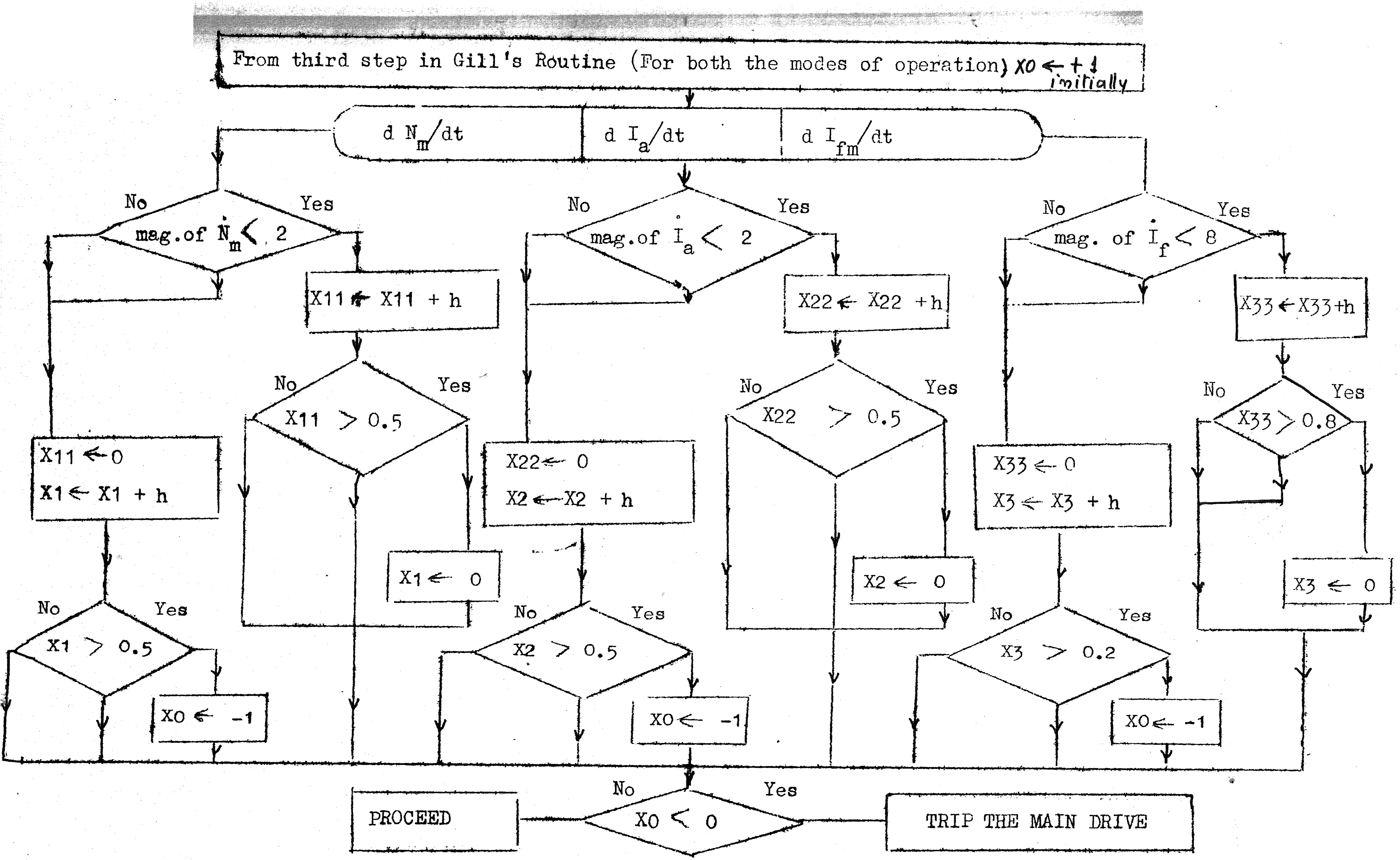


Figure 2-3

to trip).

CHAPTER - III

OPTIMISATION PROCEDURE

This chapter is devoted to a review of some known optimisation techniques associated with dynamical systems and suggests a new technique which is particularly suited to the problem stated in Chapter II. (equations 2.2.1 to 2.2.3).

The new technique suggested requires replacement of product terms by a weighted linear combination of both the terms (involved in the product) with time varying coefficients. An initial guess of the time varying coefficient may be based on an analog computer solution corresponding to an admissible control. An improvement on this approximation is obtained by taking Gateaux differential of a positive semi-definite function of state variables along the guessed trajectory. The optimal solution is obtained by using parametric expansion technique in each iteration. As a result, one solves a sequence of optimal problems converging to the required solution.

The new technique, suggested here is simple in structure and may be adapted for general trajectory manipulations required in optimisation problems.

Some theorems concerning the new technique are discussed towards the end of the chapter.

3.1 Review of Some Optimisation Techniques

The following is a brief review of some of the algorithms which have been used to solve optimisation problems. The following classification of optimisation methods has been observed in discussing these methods.

a) Gradient techniques based on iterations on boundary conditions to optimise a function of terminal state.

b) Gradient techniques for trajectory manipulations based on Taylor series expansion of optimising functional around the value in previous iteration (also called relaxation procedure).

c) Gradient techniques based on Taylor series expansion of state and adjoint variables around that of the previous iterations (also known as Generalised Newton-Raphson Method).

d) Conjugate Gradient Techniques.

e) Methods based on an approximation of optimising functional as a polynomial in state variables with time dependent coefficients (also known as Paramet ric Expansion Technique).

a) Terminal State Optimisation Procedure[2,20,22,23]

Cannonical Problem Statement :

$$\left.\begin{array}{l} \text{Given } \dot{x} = f(x, u, t) \\ \qquad x(0) = x_0 \text{ , } t \in [0, T] \\ \text{minimise } E = \phi(x(T),T) \text{ choosing } u \in U \ \forall \ t \\ x \text{ is a } n \times 1 \text{ and } u \text{ is a } m \times 1 \text{ vector} \end{array}\right\} \quad \ldots \quad \ldots \quad 3.1.1$$

The algorithm

The basis of the algorithm is often explained by associating some geometric concepts with the solution space of state equations 3.1.1. It is considered that there exists a path in solution space, emanating from the trajectory corresponding to an initially chosen admissible control, which approaches that element which minimises the functional. This path is referred to as the path of steepest descent. The algorithm, therefore, requires a sequence of control programs to be generated by calculating the tangent vector to the path of steepest descent at each step.

In general a numerical procedure will require the following computation

1) Integrate the equations of motion in forward time with an admissible control program. (If the terminal state is fixed, one may carry a measure of goodness in optimising functional detailing the difference between desired state and final state),

2) Using solutions from (1) integrate adjoint equations (3.1.2) back-ward in time

$$\left.\begin{array}{l} \dot{p} = -(\nabla_x f)/\begin{pmatrix} p \\ -1 \end{pmatrix} \\ p(T) = (\partial\phi/\partial T) \end{array}\right\} \quad .. \quad .. \quad .. \quad .. \qquad 3.1.2$$

3) Determine numerically the influence functions g_i as follows

$$g_i = (\nabla_u f)/\begin{pmatrix} p \\ -1 \end{pmatrix} \quad .. \quad .. \quad .. \quad .. \qquad 3.1.3$$

4) Modify the control program in (1) by using the following algorithm

$$\left.\begin{aligned} & u^{i+1} = u^i + \alpha (\nabla_u f) \begin{pmatrix} p \\ -1 \end{pmatrix} \\ & \text{where } \alpha \text{ is a suitable scalar} \end{aligned}\right\} \quad .. \quad .. \quad .. \quad 3.1.4$$

5) Now repeat the steps (2 to 5) until $\phi(T, x(T))$ is minimised.

Remarks

The algorithm can be used, in general, to minimise real valued convex functionals generated over normed linear spaces. When minimising functional is convex and is defined on a function space generated by system of ordinary differential equation, one can use this algorithm. The convergence of the procedure can be proved[22] when $\phi(T, x(T))$ has at least two continuous Frechet derivatives for variations in u. It may be remarked here that convergence rate slows down as the optimum is approached. In most cases a relative minimum can be achieved.

b) The Relaxation Algorithm[10,24,25]

Cannonical Problem Statement :

$$\left.\begin{aligned} & \text{Given } \dot{x} = f(x, u, t) \\ & \qquad x(0) = x_0 , \ t \in [0, T] \\ & \text{minimise } E = \int_0^T F(x, u, t)\, dt + \phi(T, x(T)) \\ & \text{subject to constraint } u \in U \quad \forall t \in [0, T] \\ & x \text{ is a } n \times 1 \text{ and } u \text{ is a } m \times 1 \text{ vector.} \end{aligned}\right\} \quad .. \quad .. \quad 3.1.5$$

The algorithm

The algorithm employs a Taylor series expansion of the return function E in each iteration. In general the approximation may contain higher

order terms, but for the description of the basic structure of the algorithm one needs to consider only up to the first order terms in the series.

The following forms the basis of the algorithm

$$E^{i+1} = \int_0^T (F^{i+1} + p^{i+1} \cdot (f^{i+1} - \dot{x}^{i+1}))\, dt + \phi^{i+1}(T, x(T)) \qquad 3.1.6^*$$

In 3.1.6 we have,

$$\left.\begin{aligned} F^{i+1} &= F^i + (\partial F^i/\partial x^i) * (x^{i+1} - x^i) + (\partial F^i/\partial u^i) * (u^{i+1} - u^i) \\ f^{i+1} &= f^i + (\partial f^i/\partial x^i) * (x^{i+1} - x^i) + (\partial f^i/\partial u^i) * (u^{i+1} - u^i) \\ p^{i+1} &= p^i + (\partial p^i/\partial x^i) * (x^{i+1} - x^i) \end{aligned}\right\} \quad 3.1.7$$

The substitution of 3.1.7 in 3.1.6 results in following approximation

$$E^{i+1} = E^i + \int_0^T \sum_{j=1}^{m} s_j^i(t) * w_j^i(t)\, dt \qquad .. \qquad .. \qquad 3.1.8$$

where

$$\left.\begin{aligned} s_j^i &= \partial F^i(t)/\partial u_j + p^i * \partial f^i/\partial u_j \quad \text{and} \\ w_j^i &= u_j^{i+1} - u_j^i \end{aligned}\right\} \qquad 3.1.9$$

The variation in control signal in equation 3.1.8 is chosen so as to ensure the condition 3.1.10

$$E^{i+1} \leq E^i \qquad .. \qquad .. \qquad .. \qquad 3.1.10$$

This leads to the following algorithm

$$u_j^{i+1} = u_j^i - \delta_j^i * \mathrm{sgn}(s_j^i) \qquad .. \qquad .. \qquad 3.1.11$$

where δ_j^i is a suitable constant consistant with the constraints of $u(t)$.

---

* Transposition is implied where ever necessary.

A typical numerical procedure would require the same steps as followed for the steepest descent algorithm except that influence functions are obtained as described in 3.1.9.

Remarks

Merriam[10] discusses equivalence of the above two approaches when only first order terms are included in the Taylor series 3.1.7. The convergence rate improves considerably when second and higher order terms are also included. Like the first method this one also can be used to optimise a real valued convex functional defined on spaces generated by ordinary differential equations.

c) Generalised Newton-Raphson Procedure[21,25,26]

Cannonical Problem Statement :

Same as in equations 3.1.5.

The algorithm

The basic principle involved is that of contraction mapping. So one attempts to obtain a transformation which will map sets of functions into improvised ones till the mapping repeats itself.

$$\left.\begin{array}{ll} \text{Define } H = F(x, u, t) + \langle p \cdot f \rangle , & \\ \text{then} \quad \dot{p} = -\partial H/\partial x , & p(T) = \partial \phi(T, x(T))/\partial x(T) \\ \quad \dot{x} = \partial H/\partial p , & x(0) = x_0 \\ \text{and } \partial H/\partial u = 0 \text{ along optimum trajectories} & \end{array}\right\} \quad .. \quad 3.1.12$$

From the condition $\partial H/\partial u = 0$ one evaluates vector u as a function

As a first step it is necessary to augment the set of adjoint equations with state equations as in equation 3.1.13

$$\left.\begin{array}{l} G(y, \dot{y}, t) = \dot{y} + g(y, t) = 0 \\ \text{where } y_1 = x_1, \ldots, y_n = x_n, y_{n+1} = p_1, \ldots, y_{2n} = p_n \end{array}\right\} \quad .. \qquad 3.1.13$$

The algorithm requires an iteration on initial values of trajectories for $y_1 \ldots\ldots y_{2n}$, which,to start with,may not correspond to an admissible control. Iterations are carried on until for further iterations the same boundary conditions are repeated.

The algorithm is derived as follows:

$$\left.\begin{array}{l} G^{i+1}(y^{i+1}, \dot{y}^{i+1}, t) = G^i(y^i, \dot{y}^i, t) + (\partial G^i/\partial y) * (y^{i+1} - y^i) \\ \qquad + (\partial G/\partial \dot{y}) * (\dot{y}^{i+1} - \dot{y}^i) = 0 \; .. \; .. \end{array}\right\} 3.1.14^*$$

From 3.1.13 it follows that $\partial G/\partial \dot{y} = 1$

On substituting this result we have the following

$$\dot{y}^{i+1} = g^i(y, t) + (\partial g^i/\partial y^i) * (y^{i+1} - y^i) \qquad .. \qquad .. \qquad 3.1.15$$

where g^i is defined in equation 3.1.13.

Equations 3.1.15 are a set of linear differential equations which can be solved by any of the conventional methods.

A numerical procedure based on Newton-Raphson iteration requires following computation :

1) Assume an arbitrary solution y(t) satisfying some boundary conditions.

---

* Transposition is implied where ever necessary

2) Calculate the partitioned transition matrix for the homogeneous linearised system corresponding to 3.1.15 by setting the following initial conditions

$$y(0) = (0, 0, \ldots, 0, y_j = 1, 0, \ldots, 0) \quad .. \quad 3.1.16$$

where j takes values from n+1 to 2n.

3) Generate a particular solution $y(t)$ of the equation 3.1.15 with initial conditions as computed in previous iteration

$$(y_1(0), y_2(0), \ldots, y_n(0), y^i_{n+1}(0), y^i_{n+2}(0), \ldots, y^i_{2n}(0)) \quad 3.1.17$$

where superscript i refers to previous iteration number. Let this solution be denoted by y_p.

4) The complete solution of the non-homogeneous system 3.1.15 is generated by a linear combination of solutions obtained in steps (1) and (2) above

$$\text{i.e.} \quad y^{i+1} = \sum_{j=n+1}^{2n} w_j * y^i_j(t) + y^i_p(t) \quad .. \quad .. \quad 3.1.18$$

where coefficients w_j are obtained by solving n algebraic equations corresponding to boundary condition at terminal time.

5) For the next iteration the initial conditions are chosen as

$$(y_1(0), y_2(0), \ldots, y_n(0), y^i_{n+1} + w_1, y^i_{n+2} + w_2, \ldots, y^i_{2n} + w_n)$$

and steps (2 to 5) are repeated until the coefficients w_j's in 3.1.18 are small and a measure corresponding to $(y^{i+1} - y^i)$ approaches zero.

Remarks

This is an indirect method unlike the two methods discussed earlier. For linear problems the convergence of the method can be tested.

The authors[26] claim advantage in terms of computation time over steepest descent method because of quadratic convergence.

d) Conjugate Gradient Technique[27,28]

Cannonical Problem Statement :

Same as in 3.1.5.

The algorithm

The basis of the algorithm is same as that for the steepest descent algorithm except that by carrying the information on gradients of two previous steps, instead of one, convergence is accelerated.

A typical numerical procedure will require following computations :

1) Integrate the equations of motion 3.1.5 in forward time with an admissible control program.

2) Using solutions as obtained in step (1) integrate adjoint equations back-ward in time

$$\left.\begin{array}{l} \dot{p}^i = \nabla_x (p^i \cdot f^i + F^i) \\ \text{with} \quad p(T) = \partial \phi(T, x(T)) / \partial x(T) \end{array}\right\} \quad .. \quad .. \qquad 3.1.20^*$$

3) Determine the gradient functional as

$$g^i = \nabla_u (\dot{p}^i \cdot f^i + F^i) \quad .. \quad .. \qquad 3.1.21$$

and from 3.1.21 generate the norm 3.1.22,

$$I^i = - \int_0^T (g^i(t) \cdot g^i(t))dt \quad .. \quad .. \quad .. \qquad 3.1.22$$

4) The new control program is obtained by following the routine given in the set of equations 3.1.23,

* Transposition is implied where ever necessary.

$$
\left.\begin{aligned}
u^{i+1} &= u^{i} + \alpha^{i} * s^{i} \\
\text{where } s^{i} &= - g^{i}(t) + \beta^{i-1} * s^{i-1} \\
\text{and } \beta^{i-1} &= (I^{i} / I^{i-1})
\end{aligned}\right\} \quad \ldots \quad \ldots \quad \ldots \qquad 3.1.23
$$

5) Repeat the steps from (2 to 5) until the trajectories are optimised.

<u>Remarks</u>

It is a direct method like steepest descent method. It may be remarked that the first step will have to be generated using steepest descent algorithm.

e) <u>Parametric Expansion Technique</u>[10,15]

Cannonical Problem Statement :

Same as in equations 3.1.5 except that

$$
E = \int_{0}^{T} F(x, u, t)dt \quad \ldots \quad \ldots \quad \ldots \quad \ldots \qquad 3.1.24
$$

<u>The algorithm</u>

Corresponding to the set of equations 3.1.5, Bellman's[2] functional equation may be written as

$$
\min_{u \in U} \left[F + \partial E/\partial t + ((\partial E/\partial x) \cdot \dot{x}) \right] = 0 \quad \ldots \quad \ldots \qquad 3.1.25
$$

The method is based on the following approximation,

$$
E = K(t) + \sum_{m=1}^{n} K_{m}(t) * x_{m}(t) + \sum_{m=1}^{n} \sum_{p=1}^{n} K_{mp} * x_{m}(t) * x_{p}(t) \qquad 3.1.26
$$

where K's refer to some auxiliary variables.

If F is quadratic expression in x and u, then it can be shown that one needs to consider only up to quadratic terms in expansion 3.1.26. This expansion was first suggested by Merriam[14]. It is easy to show that on substituting the series 3.1.26 in 3.1.25 and using the optimality condition one obtains the following

$$\left.\begin{aligned} &u = \text{a bilinear expression in } x\text{'s and } K\text{'s} \\ &\dot{K} = g(K, t) \quad \text{with} \quad K(T) = 0 \end{aligned}\right\} \quad \ldots \quad 3.1.27$$

The expression for u may be substituted in 3.1.5 so that

$$\dot{x} = f(x, K, t) \quad \ldots \quad 3.1.28$$

The flow diagram of a numerical procedure to solve optimisation problem using this technique is given in Fig.3-1.

Remarks

Systems governed by time varying differential equations can be optimised by using this method, if a finite parametric expansion for minE yields un-coupled initial value problems. The main advantage of this method is in terms of the computation time (as no iterations are involved), at the cost of additional storage. The forcing function u, as obtained in 3.1.28 generally turns out to be a continuous function over the time interval $[0, T]$.

3.2 The Proposed Trajectory Manipulation Technique

The optimisation procedures discussed in earlier section seem inadequate for the problem under consideration, because of computational difficulties involved which will be evident from the following example.

1 Formulate State model and optimality criterion

2 Generate Hamilton-Jacobi conditions for optimality using Bellman's functional equation of dynamic programming.

3 Substitute the following expression for min E in Hamilton-Jacobi equations

$$E = K_0(t) + \sum_i K_i * x_i + \sum_i \sum_j K_{ij} \quad x_i * x_j + \ldots\ldots$$

4 Obtain the structure of the control law and the Hamilton-Jacobi condition by matching coefficients.

5 Solve for auxiliary time variables back-wards in time noting that final value for all K's = 0.

6 Store the time variable trajectories corresponding to K's.

7 Solve state equations forward in time using the structure of control law and stored K trajectories.

Figure 3-1

Steps involved in the solution of optimal control problem by using Parametric Expansion method.

Example

Let it be required to minimise

$$\int_0^T c_1(x_1 - x_{1d})^2 dt \quad .. \quad .. \quad .. \quad 3.2.1$$

where x_{1d} is specified over the time $[0, T]$.

Subject to the following constraints

$$\left.\begin{aligned} \dot{x}_1 &= a_1 * x_1 + a_2 * x_2 + a_3 * x_2 * \phi(x_3) \\ \dot{x}_2 &= a_4 * x_1 + a_5 * x_2 + a_6 * x_1 * \phi(x_3) \\ \dot{x}_3 &= a_7 * x_1 + a_8 * x_2 + a_9 * x_1 * x_2 + b_{10} * u \\ u(t) &\in U \quad \forall \quad t \quad \text{in } [0, T] \end{aligned}\right\} \quad .. \quad 3.2.2$$

The corresponding set of adjoint equations are given by

$$\dot{p}_0 = 0$$

$$\left.\begin{aligned} \dot{p}_1 &= -2 * c_1 * (x_1 - x_{1d}) - a_1 - (a_4 + a_6 * \phi(x_3) - (a_7 + a_9 * x_2) \\ \dot{p}_2 &= -(a_2 + a_3 * \phi(x_3)) - a_5 * x_2 - (a_8 + a_9 * x_1) \\ \dot{p}_3 &= -a_3 * (\partial\phi(x_3)/\partial x_3) * p_1 - a_6 * (\partial\phi/\partial x_3) * p_2 \end{aligned}\right\} 3.2.3$$

The equations 3.2.2 and 3.2.3 are coupled sets of non-linear differential equations involving products of state and adjoint variables. Thus on using, steepest descent or conjugate gradient methods, one is faced with a formidable two-point boundary value problem.

One may attempt to solve this two-point boundary value problem by solving an initial value problem by using Generalised Newton-Raphson technique. The difficulty with this approach is that there is no way

of testing convergence of the algorithm for the equations of the form 3.2.2 and 3.2.3. Therefore, this method cannot be used until one has a reasonable approximation to optimal trajectories.

For the problem at hand the parametric expansion technique cannot be used, because with a finite number of terms in 3.1.26, the Bellman's functional equation is not satisfied (as the powers of x do not balance out). Infinite series approximation is not feasible computationally.

The relaxation procedure and related second variation methods of synthesis (based on Taylor series expansion) are valid over small regions of state space around optimal trajectories and hence inadequate for the problem under consideration.

The difficulties mentioned above provide sufficient motivation to evolve a new method which should be computationally feasible. The proposed method requires decomposition of original optimal control problem in two parts. The logic of this decomposition is simple : one starts with a solvable control problem, which may be based on an approximation of system modelling equations, and follow an algorithm to generate a sequence of system models whose solutions converge to the original system's solution. Since each problem in this sequence has to be solved to generate the next approximation, the original problem would be solved when a specified convergence is achieved.

The first approximation to state model is obtained on replacing all terms involving products of state variables by an approximation

of the form

$$x_i * \emptyset(x_{jg}) = w_L * x_{ig} * \emptyset(x_j) + w_K * \emptyset(x_{jg}) * x_i \quad .. \quad 3.2.4$$

where $w_L + w_K = 1$, $0 < w_L < 1$, $0 < w_K < 1$ and

and x_{ig} is an approximation to x_i

$\emptyset(x_{jg})$ is a approximation to $\emptyset(x_j)$

$\emptyset$ is a continuous one-to-one mapping.

By specifying the time trajectories x_{ig}, $\forall\ i$, which appear in the product terms, and replacing the product terms in state equations by the rule 3.2.4, one obtains essentially time varying non-linear differential equations. As an example the equations 3.2.2 may be approximated by the set 3.2.5, if $\emptyset$ is a continuous one-to-one mapping.

$$\left.\begin{aligned}
\dot{x}_1 &= a_1 * x_1 + a_2 * x_2 + 0.5 * a_3 * x_2 * \emptyset(x_{3g}) + 0.5 * a_3 * x_{2g} * \emptyset(x_3) \\
\dot{x}_2 &= a_4 * x_1 + a_5 * x_2 + 0.5 * a_6 * x_1 * \emptyset(x_{3g}) + 0.5 * a_6 * x_{1g} * \emptyset(x_3) \\
\dot{x}_3 &= a_7 * x_1 + a_8 * x_2 + 0.5 * a_9 * x_1 * x_{2g} + 0.5 * a_9 * x_{1g} * x_2
\end{aligned}\right\} \quad 3.2.5$$

With the modelling equations 3.2.5 replacing equations 3.2.2, the minimisation of the functional 3.2.1 can be performed by any of the standard optimisation procedures. However, the original problem will be solved only when it is ensured that

$$| x_{1g} - x_1 | + | x_{2g} - x_2 | + | x_{3g} - x_3 | = 0 \quad .. \quad 3.2.6$$

Thus on employing the proposed technique the following two problems have to be solved in each iteration.

Problem 1

Optimisation of performance functional E, with those equations which incorporate the latest approximation of the form 3.2.4.

Problem 2

minimise $|x_i - x_{ig}| \quad \forall \quad i$

Algorithms suited to Problem 1 have been discussed in the previous section. An algorithm for Problem 2 is given below :

1) Generate a non-negative real valued convex functional 3.2.9 $(V^k(x, x_g))$by solving equation with the approximations of previous iteration.

2) Obtain Gateaux derivative of $V^k(x, x_g)$ along each of the guessed trajectories.

3) Modify the guessed trajectories by following the rule 3.2.7

$$x_{ig}^{k+1} = x_{ig}^{k} - \epsilon * D_G V_i^k \qquad \forall \ i \qquad .. \qquad .. \qquad 3.2.7$$

where k refers to iteration number,

$D_G V_i^k$ is the Gateaux gradient functional along i-th guessed trajectory,

ϵ is a suitable constant.

4) Repeat the steps 1 through 4 until

$$\left.\begin{aligned} &D_G V_i^k = 0 \quad \forall \ i \quad \text{and} \\ &|x_{ig} - x_i| \cong 0 \quad \forall \ i \end{aligned}\right) \qquad .. \qquad .. \qquad 3.2.8$$

A flow diagram, detailing a numerical procedure, which uses parametric expansion technique for Problem 1 and the algorithm

given above for Problem 2 is given in Fig.3-2. It may be noted that the particular convex functional is generated by an expression of the form 3.2.9

$$V(t) = \int_0^t \sum_i \sum_j (x_{ig} * x_j - x_{jg} * x_i) d\sigma, \quad t \in [0, T] \qquad 3.2.9$$

Some theorems and definitions, based upon which the flow diagram in Fig.3-2 has been evolved, are discussed next.

3.3 Some Definitions and Theorems

Definition[29]-1 (Convex function) : Let X be a vector space, Y the set of real numbers then a mapping V defined on a convex subset $C \subseteq X$, to the set $B \subseteq Y$ is said to be convex if

$$\left.\begin{array}{l} V(w * x_1 + (1 - w) * x_2) \leq w * V(x_1) + (1 - w) * V(x_2) \\ \forall \quad x_1, x_2 \in C \\ \forall \quad w, 0 \leq w \leq 1 \end{array}\right\} \quad .. \qquad 3.3.1$$

Theorem - 1 : Let S be the solution set of the ordinary differential equation 3.3.2.

$$\dot{x} = f(x, x_g) \qquad .. \qquad .. \qquad .. \qquad 3.3.2$$

where $x, x_g \in C$, a convex subset of S.

Then for each $t \in [0, T]$

$$V(t, x, x_g) = \int_0^t \sum_i \sum_j (x_i * x_{jg} - x_j * x_{ig})^2 d\sigma \quad .. \qquad 3.3.3$$

is a convex real valued, non-negative functional on C.

Proof : For $x_i, x_j, x_{ig}, x_{jg} \in C$ we have

$$(x_i * x_{jg} - x_j * x_{ig})^2 \geq 0 \qquad .. \qquad .. \qquad 3.3.4$$

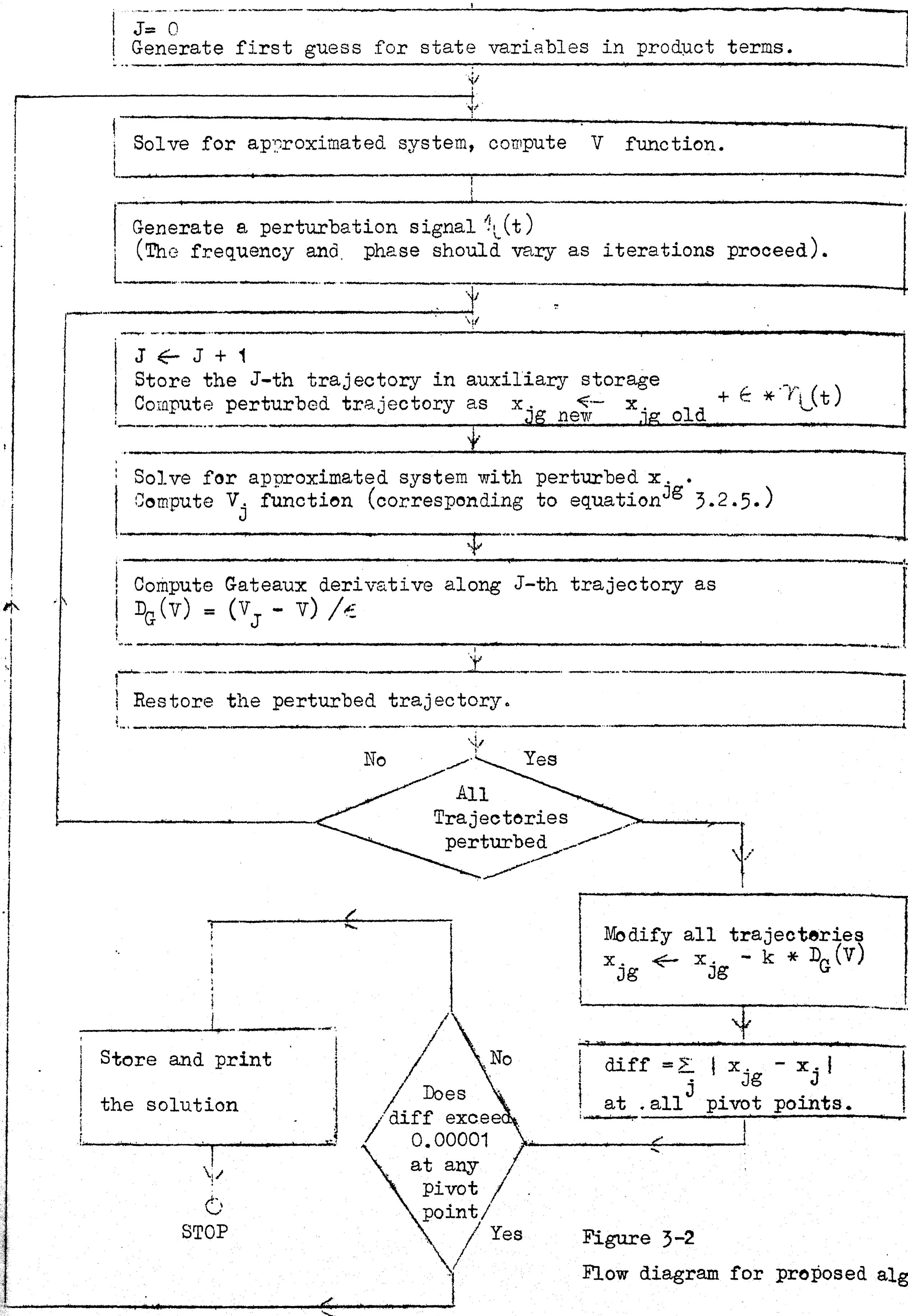


Figure 3-2

Flow diagram for proposed algorithm.

therefore $\sum_i \sum_j (x_i * x_{jg} - x_j * x_{ig})^2 \geqslant 0$ .. 3.3.5

therefore $\int_0^t \sum_i \sum_j (x_i * x_{jg} - x_j * x_{ig})^2 d\sigma \geq 0$ 3.3.6

therefore V is a real valued non-negative functional.

To prove convexity we have to show that for any $t \in [0, T]$ and for any $x_{i1}, x_{ig1}, x_{j1}, x_{jg1}, x_{i2}, x_{ig2}, x_{j2}, x_{jg2} \in C$.

$$V(t) = \int_0^t \sum_i \sum_j (w * (x_{i1} * x_{jg1} - x_{ig1} * x_{j1})$$

$$+ (1 - w) * (x_{i2} * x_{jg2} - x_{ig2} * x_{j2})^2 \quad d\sigma$$

$$\leqslant \int_0^t \sum_i \sum_j w * (x_{i1} * x_{jg1} - x_{ig1} * x_{j1})^2 d\sigma$$

$$+ \int_0^t \sum_i \sum_j (1 - w) * (x_{i2} * x_{jg2} - x_{ig2} * x_{j2})^2 d\sigma \quad 3.3.7$$

To prove inequality 3.3.7, we proceed as follows :

Let $x_{ik} * x_{jgk} - x_{igk} * x_{jk} = a_{ijk}$ 3.3.8

if $a_{ij2} \leqslant a_{ij1}$ then for $0 \leqslant w \leqslant 1$

$w * (a_{ij1} - a_{ij2}) \leqslant a_{ij1} - a_{ij2}$

or $w * (a_{ij1} - a_{ij2}) + 2 * a_{ij2} \leqslant a_{ij1} + a_{ij2}$

or $w^2 * (a_{ij1} - a_{ij2})^2 + 2 * w * a_{ij2} * (a_{ij1} - a_{ij2})$

$$\leqslant w * (a_{ij1}^2 - a_{ij2}^2)$$

or $(w * (a_{ij1} - a_{ij2}) + a_{ij2})^2 \leqslant w * a_{ij1}^2 - w * a_{ij2}^2 + a_{ij2}^2$

or $(w * a_{ij1} + (1 - w) * a_{ij2})^2 \leqslant w * a_{ij1}^2 + (1 - w) * a_{ij2}^2$ 3.3.9

If $a_{ij2} \geqslant a_{ij1}$ one could start with the inequality

$$(1 - w) * (a_{ij2} - a_{ij1}) \leqslant (a_{ij2} - a_{ij1}) \quad .. \quad .. \quad 3.3.10$$

and arrive at the result 3.3.9.

From 3.3.9 it follows that for each $t \in [0, T]$

$$\int_0^t \sum_i \sum_j (w * a_{ij1} + (1 - w) * a_{ij2})^2) d\sigma \leqslant \int_0^t \sum_i \sum_j w * (a_{ij1})^2 d\sigma$$
$$+ \int_0^t \sum_i \sum_j (1 - w) * (a_{ij2})^2 d\sigma .. \qquad 3.3.11$$

On substituting the values for a_{ij1} and a_{ij2} from 3.3.8 we have from 3.3.11

$$\int_0^t \sum_i \sum_j (w * (x_{i1} * x_{ig1} - x_{ig1} * x_{j1}) + (1 - w) * (x_{i2} * x_{ig2} - x_{ig2} * x_{j2}))d\sigma$$
$$\leqslant \int_0^t \sum_j \sum_i w * (x_{i1} * x_{ig1} - x_{ig1} * x_{j1})^2 d\sigma$$
$$+ \int_0^t \sum_i \sum_j (1 - w) * (x_{i2} * x_{ig2} - x_{ij2} * x_{j2})^2 d\sigma \quad 3.3.12$$

The inequality 3.3.12 establishes convexity of the functional V for each t.

Remark - 1 : V is a continuous function on C for each t.

Remark - 2 : V has a lower bound equal to zero and V = 0 when either $x_{ik} = x_{igk}$ $\forall$ i, for all k or $x_{igk} = 0$ $\forall$ i $\forall$ k

Definition - 2 (<u>Gateaux differential, Gateaux differentiable, Gateaux derivative</u>) : Let X be a vector space, Y a normed space, and V a mapping defined on a domain $A \subseteq X$, to the set $B \subseteq Y$. Then

for any $x \in A$ and arbitrary $h \in X$, the limit 3.3.13,

$$\left.\begin{array}{l} \lim\limits_{a \to 0} \left[V(x + a * h) - V(x) \right] / a \\ \text{where 'a' is a scalar} \end{array}\right\} \qquad \ldots \qquad \ldots \qquad 3.3.13$$

if it exists, is known, as <u>Gateaux differential of V</u> at x with increment h, and is denoted as $D_G V(x, h)$.

Further if the limit exists $\forall$ $h \in X$, the mapping V is said to be <u>Gateaux differentiable at x</u>.

If V is Gateaux differentiable then <u>Gateaux derivative of V</u> at the point X is defined to be the mapping $V'_x : X \to Y$ given by

$$V'_x(h) = D_G(V(x, h)) \qquad \ldots \qquad \ldots \qquad 3.3.14$$

Remark - 1 : When Y is the real line, then Gateaux differential of V at x with increment h, if it exists, is given by

$$D_G V(x, h) = d\ (V(x + a * h))/da \Big|_{a=0} \qquad \ldots \qquad 3.3.15$$

Remark - 2 : There are three terms defined above

1) Gateaux differential

2) Gateaux differentiable

3) Gateaux derivative

Each term has to be used in appropriate context.

Theorem - 2 : Let V be a real valued Gateaux differentiable function on a vector space X. A necessary condition for V to have an extremum at $x_0 \in X$ is that $D_G V(x, h) = 0$.

Proof : As V is Gateaux differentiable and has an extremum at $x_0 \in X$, it follows from ordinary calculus, that $\forall \; h \in X$

$$(d\,(V(x + a * h))/da)\Big|_{a=0} = 0 \qquad .. \qquad .. \qquad 3.3.16$$

Remark - 1 : Let S be the solution set of an ordinary differential equation 3.3.17,

$$\dot{x} = f(x, x_g) \qquad .. \qquad .. \qquad 3.3.17$$

V a convex real valued Gateaux differentiable functional defined on a convex subset $C \subseteq S$ then a necessary condition for V to have an extremum at $x_0 \in C$ is that $D_G V(x, h) = 0$, $\forall \; h \in X$.

Theorem - 3 : For the conditions given in Theorem - 1, the function V is Gateaux differentiable.

Proof : To show that V is Gateaux differentiable, it is sufficient to show that the limit of the integrand in 3.3 exists, for each $t \in [0, T]$, for arbitrary $h \in S$.

Let 'a' be any scalar then for any $h_i, h_g \in S$ we consider the following limit :

$$\lim_{a \to 0} \sum_i \sum_j \Big[\big[((x_i + a * h_i) * (x_{jg} + a * h_{jg}) - (x_j + a * h_j) * (x_{ig} + a * h_{ig}))^2 - (x_i * x_{jg} - x_j * x_{ig})^2 \big] / a \Big] \quad 3.3.18$$

$$= \sum_i \sum_j \lim_{a \to 0} \Big[\big[((x_i * x_{jg} + a^2 * h_i * h_{jg} + a * h_i * x_{jg} + a * h_{jg} * x_i - x_j * x_{ig} - a^2 * h_{ig} * h_j - a * h_{ig} * x_j - a * h_j * x_{ig}))^2 - (x_i * x_{jg} - x_j * x_{ig})^2 \big] / a \Big] \quad 3.3.19$$

Let $x_i * x_{jg} - x_j * x_{ig} = q_{1ij}$, $h_i * h_{jg} - h_{ig} * h_j = q_{2ij}$

$h_i * x_{jg} - h_j * x_{ig} = q_{3ij}$, $h_{jg} * x_i - h_{ig} * x_j = q_{4ij}$

Then 3.3.19 shall be

$$= \sum_i \sum_j \lim_{a \to 0} \left[\left[(q_{1ij} + a^2 * q_{2ij} + a * (q_{3ij} + q_{4ij}))^2 - q_{1ij}^2 \right] / a \right] \quad .. \quad .. \quad 3.3.20a$$

$$= \sum_i \sum_j (q_{3ij} + q_{4ij}) * q_{1ij}$$

$$= \sum_i \sum_j (h_i * x_{jg} - h_j * x_{ig} + h_{jg} * x_i - h_{ig} * x_j) * (x_i * x_{jg} - x_j * x_{ig}) \quad .. \quad .. \quad 3.3.20b$$

Expression 3.3.20b demonstrates that the limit exists $\forall \; h \in X$.

Remark - 1 : In Theorem - 3 $D_G V(x, h) = 0$ when $h_i = x_i$, $h_{ig} = x_{ig}$.

This is obvious from 3.3.20b.

Remark - 2 : In Theorem - 3 $D_G V(x, h) = 0$ for all $h \in S$, when

$x_i = x_{ig} \quad \forall \; i$.

This is a consequence of Theorem - 2. The lower bound of the function V is zero. The function V assumes a value equal to zero when $x_i = x_{ig}$.

Theorem - 4 : Let S be the solution set of the ordinary differential equation 3.3.17, V a continuous, real valued, convex functional defined on a convex subset $C \subseteq S$. Then

$$x_g^i = (x_{1g1}^i, x_{1g2}^i, \ldots, x_{1gn}^i) \in C \qquad .. \qquad .. \qquad 3.3.21$$

implies

$$x_g^{i+1} = x_g^i - \nabla_G V(x_g^i, h) * a * h \in C \qquad .. \qquad .. \qquad 3.3.22$$

where 1) $h = (h_1, h_2, \ldots, h_n)$

2) $\nabla_G V(x_g^i, h) * a * h = (D_G V(x_{1g1}^i, h_1) * a * h_1, \ldots$

$\ldots, D_G V(x_{1gn}^i, h_n) * a * h_n)$

for any real 'a' satisfying the inequality 3.3.23

$$0 \leqslant a \| \nabla_G V(x_g^i, h) * h \| \leqslant \delta(\epsilon) \qquad .. \qquad .. \qquad 3.3.24$$

where $\delta(\epsilon)$ refers to a neighbourhood of x_g^i for a corresponding ϵ neighbourhood of $V(x_g^i)$ with $\epsilon > 0$ as determined by continuity of V in x_g^i.

Proof : Let $x_g^i \in C$, V is a continuous function of x_g^i defined on C.

For any $x_g^i \in C$ and $\epsilon > 0$ $\quad V(x_g^{i+1}) \in N(V(x_g^i), \epsilon)$

We have $V(x_g^{i+1}) = V(x_g^i) - \nabla V * (x_g^{i+1} - x_g^i) + 0 \ldots \qquad .. \qquad 3.3.25$

where $\nabla V = (\partial V/\partial x_{1g1}^i, \partial V/\partial x_{1g2}^i, \ldots, \partial V/\partial x_{1gn}^i) \quad .. \qquad 3.3.26$

and $x_g^{i+1} \in N(x_g^i, \delta(\epsilon)) \qquad .. \qquad .. \qquad 3.3.27$

or $\| x_g^{i+1} - x_g^i \| \leqslant \delta(\epsilon) \qquad .. \qquad .. \qquad 3.3.28$

So now if one follows the rule 3.3.22 ensuring 3.3.29

$$\| D_G V(x_{1g1}^i, h_1) * a * h_1, \ldots, D_G V(x_{1gn}^i, h_n) * a * h_n \| \leqslant \delta(\epsilon) \quad 3.3.29$$

Then $x_g^{i+1} \in C$.

It follows from 3.3.29 that the scalar $a > 0$ is to be chosen ensuring 3.3.30.

$$0 \leqslant a * \| \nabla_G V(x_g^i, h) * h \| \leqslant \delta(\epsilon) \qquad .. \qquad .. \qquad 3.3.30$$

where $\delta(\epsilon)$ refers to a neighbourhood of x_g^i for a corresponding ϵ neighbourhood of $V(x_g^i)$ with $\epsilon > 0$ as determined by continuity of V.

Theorem - 5 : Given the conditions of Theorem - 3, a necessary condition for the sequence x_{jg}^k to converge to some $\overset{o}{x}_{jg}$ is that for a fixed scalar 'a' and increment h

$$\| \nabla G V^{k+1} \| \leqslant \| \nabla G V^k \|, \qquad .. \qquad .. \qquad 3.3.31$$

where k refers to iteration number.

This result is an immediate consequence of previous theorems.

Remark - 1 : The process of convergence will slow down as iterations proceed, because for fixed 'a' and 'h' we notice that step sizes reduce as iterations proceed.

Remark - 2 : For the V function defined in 3.3.3 the sequence will converge when

$$x_i = x_{ig} \quad \forall \; i \qquad 3.3.32$$

as this is the condition when $\nabla_G V = 0$.

CHAPTER - IV

THE CONTROL OPERATIONS ON MAIN DRIVE

In the previous two chapters the modelling and optimisation procedures were discussed. In this chapter the problem stated in Chapter II will be solved by using the optimisation techniques discussed in Chapter III.

It may be recalled that during the generator field control mode of operation the state equations are linear and during the motor field control mode of operation, the state equations are non-linear and involve product and functional types of non-linearities. For both the sets of equations parametric expansion technique has been used to obtain the control program. The use of iterative trajectory manipulation technique described in Chapter III is demonstrated in computation of a control law for motor field control mode of operation.

This chapter includes discussion of computational aspects like step size selection and selection of integration procedure etc. The results of computation are given in various graphs and tables.

The chapter ends with a description of a hypothetical situation detailing computer aided execution of optimal control program for operation of all speed ranges of the main drive.

4.1 Speed Control by Generator Field Voltage

The problem of speed control during generator field control mode of operation can be stated as follows :

Given the following state equations

$$
\left.
\begin{aligned}
&d\, I_{fg}/dt = a_{11} * I_{fg} + b_1 * V_{fg} \\
&d\, I_a/dt = a_{21} * I_{fg} + a_{22} * I_a + a_{24} * N_m * \phi(I_{fm}) \\
&d\, N_m/dt = a_{42} * \phi(I_{fm}) * I_a + b_2 * T_L \\
&d\, I_{fm}/dt = 0 \quad , \quad \phi(I_{fm}) = 1.0 \text{ p.u.} \\
&\text{where} \quad a_{11} = -b_1 = -0.288 \\
&\qquad a_{21} = -a_{22} = -a_{24} = 11.09 \\
&\qquad a_{42} = b_2 = 2.24
\end{aligned}
\right) \quad .. \quad 4.1.1
$$

Find V_{fg} to minimise

$$E = \int_0^T \left[A * (N_m - N_{md})^2 + B * V_{fg} + V_{fg}^2 + D * (V_t - V_{td})^2 \right] dt \quad 4.1.2$$

where V_t and V_{td} refer to actual and desired terminal voltages.

$V_t = K_g * I_{fg} - r_{ag} * I_a$

$V_{td} = 1$ when rolling takes place at top speed

$D = 0$ when reversals of speed take place

The magnitudes of A, B and D are obtained by following the logic of iteration discussed in Chapter II, so that, the following conditions are ensured :

1) The maximum magnitude of V_{fg} does not exceed 10.0 p.u.

2) The form of V_{fg} is continuous.

3) The performance specifications discussed in Chapter II are achieved.

The Hamilton-Jacobi equations corresponding to this problem can be written as

$$\min_{V_{fg} \in V} \Big[A * (N_m - N_{md})^2 + B * V_{fg} + D * (V_t - V_{td})^2 + \partial E/\partial t + (\partial E/\partial I_{fg}) * \dot{I}_{fg} + (\partial E/\partial I_a) * \dot{I}_a + (\partial E/\partial N_m) * \dot{N}_m \Big] \qquad 4.1.3$$

Let $E = K_1(t) + K_2(t) * I_{fg} + K_3(t) * I_a + K_4(t) * N_m + K_5(t) * I_{fg}^2 + K_6(t) * I_{fg} * I_a + K_7(t) * I_{fg} * N_m + K_8(t) * I_a^2 + K_9(t) * I_a * N_m + K_{10}(t) * N_m^2 \quad \ldots \quad \ldots \quad \ldots \qquad 4.1.4$

On substituting the expression 4.1.4 and equations 4.1.1 in 4.1.3 one obtains the control law 4.1.5 and auxiliary differential equations 4.1.6 as follows:

$$V_{fg} = - b_1 * (K_2 + 2 * K_5 * I_{fg} + K_6 * I_a + K_7 * N_m) - B \qquad 4.1.5$$

$$d\, K_1/dt + K_4 * b_2 * T_L + A * N_{md}^2 - 0.5 * (K_2 * b_1 + B)^2 = 0$$

$$d\, K_2/dt + K_3 * a_{21} + K_7 * b_2 * T_L + K_2 * a_{11} - 2 * K_5 * b_1 * (K_2 * b_1 + B) = 0$$

$$d\, K_3/dt + K_3 * a_{22} + K_4 * a_{42} * \phi(I_{fm}) + K_9 * b_2 * T_L - K_6 * b_1 * (K_2 * b_1 + B) = 0$$

$$d\, K_4/dt + K_3 * a_{24} * \phi(I_{fm}) + 2 * K_{10} * b_2 * T_L - 2 * A * N_{md} - K_7 * b_1 * (K_2 * b_1 + B) = 0$$

$$d\, K_5/dt + 2 * K_5 * a_{11} + K_6 * a_{21} - 2 * K_5^2 * b_1^2 = 0$$

$$d\, K_6/dt + K_6 * a_{11} + K_6 * a_{22} + 2 * K_8 * a_{21} + K_7 * a_{42} * \phi(I_{fm}) - 2 * K_5 * K_6 * b_1^2 = 0$$

$$\left.\begin{aligned}
&d\,K_7/dt + K_7 * a_{11} + K_6 * a_{24} * \phi(I_{fm}) + K_9 * a_{21} \\
&\qquad - 2 * K_5 * K_7 * b_1^2 = 0 \\
&d\,K_8/dt + 2 * K_8 * a_{24} * \phi(I_{fm}) + K_9 * a_{42} * \phi(I_{fm}) \\
&\qquad - 0.5 * K_6^2 * b_1^2 = 0 \\
&d\,K_9/dt + 2 * K_8 * a_{24} * \phi(I_{fm}) + K_9 * a_{22} + 2 * K_{10} * \\
&\qquad a_{42} * \phi(I_{fm}) - K_6 * K_7 * b_1^2 = 0 \\
&d\,K_{10}/dt + K_9 * a_{24} * \phi(I_{fm}) + A - 0.5 * K_7^2 * b_1^2 = 0
\end{aligned}\right\} \quad 4.1.6$$

$$K_1(T) = K_2(T) = \ldots = K_{10}(T) = 0$$

To obtain the optimal solution one first solves the auxiliary differential equations 4.1.6 back-wards in time and stores those trajectories which are used in obtaining optimal control law. Now using the control law 4.1.5 one solves the state equations 4.1.1 in forward time to yield optimal state trajectories. The discussion on computer results is given in Section 4.5 after evolving the computational strategy for the motor field control mode of speed control.

4.2 Speed Control by Motor Field Voltage

The problem of speed control during motor field control mode of operation can be stated as follows :

Given the following state equations

$$\left.\begin{aligned}
&d\,I_{fg}/dt = 0 \\
&d\,I_a/dt = a_{21} * I_{fg} + a_{22} * I_a + a_{24} * N_m * \phi(I_{fm})
\end{aligned}\right\}$$

$$\left.\begin{aligned}
d\,I_{fm}/dt &= a_{33} * K_{fm} * I_{fm} + b_3 * K_{fm} * V_{fm} \\
d\,N_m/dt &= a_{42} * \phi(I_{fm}) * I_a + b_2 * T_L \\
\phi(I_{fm}) &= I_{fm} \quad \text{when } I_{fm} \leqslant 0.75 \text{ p.u.} \\
&= 0.75 + 0.8 * (I_{fm} - 0.75) \\
&= 0.95 + 0.1515 * (I_{fm} - 1.0)
\end{aligned}\right) \quad 4.2.1$$

$$K_{fm} = \triangle I_{fm} / \triangle \phi(I_{fm}) \quad \ldots \quad \ldots \text{ (adjusts field time constant)}$$

$$a_{33} = -b_3 = -0.218$$

Find V_{fm} to minimise

$$E = \int_0^T \left[A * (N_m - N_{md})^2 + B * V_{fm} + V_{fm}^2 + D * (V_t - V_{td})^2 \right] dt \qquad 4.2.2$$

where V_t and V_{td} refer to actual and desired terminal voltages.

$$V_t = \phi(I_{fm}) * N_m + I_a * r_{am}$$

$$V_{td} = 1.0 \text{ p.u.}$$

The magnitudes of A, B and D are obtained by following the logic of iteration discussed in Chapter II, so that the following conditions are ensured.

1) The maximum magnitude of V_{fm} does not exceed 10 p.u.

2) The form of V_{fm} is continuous.

3) The desired performance specifications as stated in Chapter II are achieved.

The Hamilton-Jacobi equations corresponding to the above problem

can be written as

$$\min_{V_{fm} \in V} \left[A * (N_m - N_{md})^2 + B * V_{fm} + V_{fm}^2 + D * (V_t - V_{td})^2 + \partial E/\partial t + (\partial E/\partial I_{fm}) * \dot{I}_{fm} + (\partial E/\partial I_a) * \dot{I}_a + (\partial E/\partial N_m) * \dot{N}_m \right] = 0 \quad 4.2.3$$

To use the iterative trajectory manipulation technique, the system equations are first approximated to the form

$$\left.\begin{aligned} d\, I_{fg}/dt &= 0 \\ d\, I_a/dt &= a_{21} * I_{fg} + a_{22} * I_a + 0.5 * a'_{24} * (N_{mg} \quad I_{fm} + I_{fmg} \quad N_m) \\ d\, I_{fm}/dt &= a_{33} * K_{fm} * I_{fm} + b_3 * K_{fm} * V_{fm} \\ d\, N_m/dt &= 0.5 * a'_{42} * I_{fmg} * I_a + 0.5 * a'_{42} * I_{fm} * I_{ag} + b_2 * T_L \end{aligned}\right\} \quad 4.2.4$$

where a'_{24}, a'_{42} are modified coefficients corresponding to a_{24}, a_{42} to have piece-wise linear approximation to $\phi(I_{fm})$ versus I_{fm} characteristic.

I_{fmg}, I_{ag} and N_{mg} correspond to the guesses of the solution I_{fm}, I_a and N_m respectively. As a first guess I_{fmg}, I_{ag} and N_{mg} are generated from the analog computer solutions discussed in Appendix A.

$$\text{Let } E = K_1 + K_2 * I_a + K_3 * I_{fm} + K_4 * N_m + K_5 * I_a^2 + K_6 * I_a * I_{fm} + K_7 * I_a * N_m + K_8 * I_{fm}^2 + K_9 * I_{fm} * N_m + K_{10} * N_m^2 \quad .. \quad 4.2.5$$

On substituting equations 4.2.4 and 4.2.5 in 4.2.3 and applying optimality conditions one obtains the control law and auxiliary differential equations as follows

$$V_{fm} = - K_{fm} * b_3 * (K_3 + K_6 * I_a + K_9 * N_m + 2 * K_8 * I_{fm} + B / b_3) \quad 4.2.6$$

$$
\left.
\begin{aligned}
& d\,K_1/dt + K_2 * a_{21} * I_{fg} + K_4 * b_2 * T_L + A * N_{md}^2 - 0.5 * \\
& \qquad (K_3 * b_3 * K_{fm} + B)^2 + 2.5 * D = 0 \\
& d\,K_2/dt + K_2 * a_{22} + 2 * K_5 * I_{fg} + K_7 * b_2 * T_L - K_6 * b_3 * K_{fm} * \\
& \qquad (K_3 * b_3 * K_{fm} + B)^2 + (0.5 * K_4 - D) * a'_{42} * I_{fmg} = 0 \\
& d\,K_3/dt + 0.5 * (K_2 * a'_{24} * N_{mg} + K_4 * a'_{42} * I_{ag}) + K_3 * a_{33} * K_{fm} \\
& \qquad - 2 * K_8 * (K_3 * b_3 * K_{fm} + B) * b_3 * K_{fm} + K_9 * b_2 * T_L \\
& \qquad + K_6 * a_{21} * I_{fg} - D * (I_{ag} * a'_{42} + 2 * U_{fm} * N_{mg}) = 0 \\
& d\,K_4/dt + a_{21} * K_7 * I_{fg} + 0.5 * K_2 * a'_{24} * I_{fmg} + 2 * (- A * N_{md} \\
& \qquad + K_{10} * b_2 * T_L - U_{fm} * D * I_{fmg} + 0.5 * K_9 * b_3 * K_{fm} * \\
& \qquad (K_3 * b_3 * K_{fm} + B)) = 0 \\
& d\,K_5/dt + 2 * K_5 * a_{22} - 0.5 * ((K_6 * b_3 * K_{fm})^2 - K_7 * a'_{42} * I_{fmg}) \\
& \qquad + D * (a'_{42} * I_{fmg})^2 = 0 \\
& d\,K_6/dt + K_5 * a'_{24} * N_{mg} + K_6 * (a'_{42} + a_{33} * K_{fm}) + 0.5 * a'_{42} * \\
& \qquad (K_7 * I_{ag} + K_9 * I_{fmg}) - 2 * K_8 * K_6 * (b_3 * K_{fm})^2 = 0 \\
& d\,K_7/dt + K_7 * a'_{42} - K_6 * K_9 * (b_3 * K_{fm})^2 + (K_5 * a'_{24} \\
& \qquad + K_{10} * a'_{42}) * I_{fmg} = 0 \\
& d\,K_8/dt + 0.5 * (K_6 * a'_{24} * N_{mg} + K_9 * a'_{42} * I_{ag}) + D * ((a'_{42} * I_a)^2 \\
& \qquad + (U_{fm} * N_{mg})^2) + 2 * K_8 * K_{fm} * (a_{33} - K_8 * b_3^2 * K_{fm}) = 0
\end{aligned}
\right\}
$$

$$\left.\begin{aligned}
&d\,K_9/dt + 0.5 * a'_{24} * (K_7 * N_{mg} + K_6 * I_{fmg}) + K_{10} * a'_{42} * I_{ag} \\
&\qquad - 2 * K_8 * K_9 * (b_3 * K_{fm})^2 + K_9 * a_{33} * K_{fm} = 0 \\
&d\,K_{10}/dt + A + U_{fm}^2 * (0.5 * K_9^2 * b_3^2 + D * I_{fmg}^2) \\
&\qquad + 0.5 * K_7 * a'_{24} * I_{fmg} = 0
\end{aligned}\right\} \quad 4.2.7$$

$$K_1(T) = K_2(T) = \ldots = K_{10}(T) = 0$$

where U_{fm} is used as a coefficient to compute terminal voltage (a function of $\phi(I_{fm})$) in terms of I_{fm}.

To obtain the optimal solution to the original problem (corresponding to equations 4.2.1 and 4.2.2) one first solves the optimal control problem corresponding to equations 4.2.4 and improves upon these solutions till the solutions corresponding to the original problem are obtained.

The auxiliary differential equations 4.2.7 (corresponding to the approximation 4.2.4) are solved back-wards in time and those trajectories which are used in obtaining optimal control law are stored. This storage is in addition to the storage required for the approximating trajectories (I_{fmg}, N_{mg} and I_{ag}). Using the control law 4.2.5, one solves the state equations 4.2.4 in forward time and also generates a convex real valued non-negative functional corresponding to the expression 4.2.8 (notice the similarity with the expression 3.3.3)

$$V = \int_0^t \Big[(I_{fmg} * I_a - I_{ag} * I_{fm})^2 + (I_a * N_{mg} - I_{ag} * N_m)^2 + (I_{fm} * N_{mg} - I_{fmg} * N_m)^2 \Big] d\tau \quad \ldots \quad 4.2.8$$

To improve upon this approximation it is required to evaluate Gateaux Differential along the trajectories I_{ag}, I_{fmg} and N_{mg}.

To evaluate Gateaux differential, one first obtains a solution of the set 4.2.1 for some continuous form of V_{fm} (preferably with oscillations), and use these solutions to generate variations on each of the trajectories of I_{ag}, I_{fmg} and N_{mg}. For this computation a sinusoidal perturbation over-riding an admissible control was used to generate variations to be used in evaluating Gateaux differential. The frequency of the sinusoidal perturbation was varied as the iterations proceeded. The numerical procedure detailing the evaluation of Gateaux differential is discussed later in this chapter.

The trajectories are modified next by following the rule 4.2.9 :

$$\left.\begin{aligned} N_{mg\ new} &= N_{mg\ old} - D_G V(N_{mg\ old}, N_{mp}) \\ I_{ag\ new} &= I_{ag\ old} - D_G V(I_{ag\ old}, I_{ap}) \\ I_{mg\ new} &= I_{fm\ old} - D_G V(I_{fmg\ old}, I_{fmp}) \end{aligned}\right) \quad .. \qquad 4.2.9$$

where the subscript p refers to the solutions obtained with a perturbation riding over an admissible control to determine the general direction of improvement for the guesses I_{fmg}, I_{ag} and N_{mg}.

This modification scheme is carried on till

$$\left| N_m - N_{mg} \right| + \left| I_a - I_{ag} \right| + \left| I_{fm} - I_{fmg} \right| \leqslant 0.0001 \qquad 4.2.10$$

which ensures that the original problem is solved optimally if

$$\nabla G^V(I_{fm}, I_a, N_m, I_{fmg}, I_{ag}, N_{mg}, I_{fmp}, I_{ap}, N_{mp}) = 0 \ .. \qquad 4.2.11$$

4.3 Computational Considerations

Selection of step size :

The optimisation algorithm employs parametric expansion technique. So solutions to the auxiliary time variables are first obtained back-wards in time and these trajectories are stored to be used in obtaining forward time solution of the state variables later. It is obvious that if the initial step sizes (corresponding to the terminal time in (0, T))are small, then error propagation towards the end (corresponding to initial time in (0, T)) would be small. Therefore the computational grid near terminal time must be very fine.

The analog computer solutions (feasible solutions discussed in Appendix A) indicate high rate of change in state variable trajectories during initial portions in time interval (0, T). This would suggest that step sizes near the initial time must be small to obtain the required numerical accuracy in the solutions of state variables.

(For all computations here, the required numerical accuracy was placed at eight significant digits.)

The arguments above suggest smaller interval size at both the ends of every time slice (0, T). For this computation every time slice was divided in 500 intervals such that the step size decreased linearly towards the ends as described by the expression 4.3.1.

$$h_n = h_{n-1} + 0.00001450 * ((250.1 - n) / (250.1 - n)) * T \qquad .. 4.3.1$$

where n refers to the number of interval and $h_1 = T / 5000$.

One can, in general, generate an error bound[33] controlled step size, however, for operational considerations a fixed routine, like 4.3.1, is to be preferred. Besides this, a strong justification for using the expression 4.3.1 lies in the fact that on halving the step sizes no particular improvement in eighth significant digit is observed.

Selection of method :

The variable step size restricts the choice of solution methods to Runge-Kutta type routines. Gill's[19] modified version of fourth-order Runge-Kutta method was used in the present computation. The round off errors accumulated at each step were found to be smaller than eights significant digit. The smallness of round off errors at any step can be ascertained by following the procedure suggested by Gill. The procedure requires a comparison of the results when the intermediate variables are stored in double precision with those obtained when these variables are stored in single precision. If the two results compare well then the smallness of round off error in every step and overall numerical stability both are ensured.

4.4 The Computer Program

The complete flow diagram is divided in four parts so that different aspects of the program can be emphasized evenly.

Fig. 4-1 : The flow diagram in Fig.4-1 essentially depicts the rolling schedule. The switches F, FF, FFF, FFFF, GG, GGG, GGGG and BF combine to yield different modes of operation.

Switch F* : +1 for generator field control mode of operation.

-1 for motor field control mode of operation.

Switch FF : +1 for rolling at top speed of operation.

-1 for rolling below and at base speed of operation.

Switch FFF : +1 for - base speed to + base speed of operation.

-1 for + base speed to - base speed of operation.

$$N_{md}(K) = - FFF * (2 - (\sum_{1}^{K} h_i) / T) \quad .. \qquad 4.4.1$$

Switch FFFF : +1 for + top speed of operation.

-1 for - top speed of operation.

$$N_{md}(K) = FFFF * 2 \quad .. \qquad 4.4.2$$

Switch BF : +1 for loading at top speed of operation.

-1 for load throw-off at top speed of operation.

Switch GG : +1 for field weakening condition (to race to top speed from base speed).

-1 for field strengthening condition (for changes in speed from top speed to base speed).

Switch GGG : +1 for + base speed to + top speed of operation.

-1 for - base speed to - top speed of operation.

$$N_{md}(K) = GGG * (1 + (\sum_{1}^{K} h_i) / T) \quad .. \qquad 4.4.3$$

Switch GGGG : +1 for + top speed to + base speed of operation.

-1 for - top speed to - base speed of operation.

$$N_{md}(K) = GGGG * (2 - (\sum_{1}^{K} h_i) / T) \quad .. \qquad 4.4.4$$

In operational stage these switch combinations will generate the scheduling algorithm for desired speed of operation.

---

* The two switch positions are denoted as +1 and -1.

Fig. 4-2 : The flow diagram in this figure refers to common program which is used during both the modes of operation. The flow diagram, as it is drawn, assumes switch position F as -1 i.e., the motor field control mode. When the switch position is +1 the function generation routine is skipped.

The phrase "for all J" refers to a DO loop in FORTRAN PROGRAM. Similarly h_n refers to n-th interval.

Various coefficients used in generating intermediate steps in Gill's routine in fact can be generated once and stored to be used in computing intermediate steps in solution of the auxiliary equations as well as the state equations.

During operational stage one need not store the matrix K(J, N). One will only store those trajectories which are used in generating control law as this is the only useful information during a particular time slice.

Fig. 4-3 : The flow diagram in this figure refers to the common program which is used during both the modes of operation. The flow diagram is drawn with the assumption that switch F is in -1 position. Whenever switch F is in +1 position the function generation routine in the flow diagram is skipped.

A particular feature of this flow diagram is the simulation of protection clocks. The flow diagram for the clock incrementing routine is given in Fig.2-3. Here the clock routine is simulated

Perturbation :

The perturbations are generated from the solution of equation 4.4.5

$$\left.\begin{aligned} \dot{x}^{k+1} &= f(x^{k+1}, u^{k+1}, t) \\ \text{where } u^{k+1} &= u^{k} + \mu * \eta(k) \\ \eta &= \operatorname{Sin} 0.1 * k * t \\ \mu &= 0.005 \end{aligned}\right) \quad .. \quad 4.4.5$$

This perturbation is used as the increment 'a * h' as described in Section 3.3.

Gateaux differential :

To generate the Gateaux differential for each n (n takes the values from 1 to 500) on each of the trajectories of I_{fmg}, I_{ag} and N_{mg} the following steps are followed :

1) Store the trajectory (say I_{fmg}) in auxiliary storage.

2) For n = 1 through 500, generate

$$I_{fmg\ new} = I_{fmg\ old} + 0.005 \ * \text{(corresponding trajectory in 4.4.5)} \qquad 4.4.6$$

3) Corresponding to the change 4.4.6 compute $I_{fm\ new}$, $N_{m\ new}$ and $I_{a\ new}$ and simultaneously generate $V_{+1}(t)$

$$\begin{aligned} V_{+1}(t) = \int_0^t \Big[& (I_{a\ new} * I_{fmg\ new} - I_{fm\ new} * I_{ag\ old})^2 + \\ & (I_{ag\ old} * N_{m\ new} - N_{mg\ old} * I_{a} \quad)^2 + \\ & (I_{fmg\ new} * N_{m\ new} - N_{mg\ old} * I_{fm\ new})^2 \Big] d\sigma \end{aligned} \qquad 4.4.7$$

by summing the corresponding stair case approximation of terms in 4.4.7.

4) For n = 1 through 500 generate 4.4.8.

$$I_{fmg\ new} = I_{fmg\ old} - 0.005 * (\text{corresponding solution in } 4.4.5) \qquad 4.4.8$$

5) Corresponding the change 4.4.8 compute $I_{fm\ new}$, $N_{m\ new}$ and $I_{a\ new}$ and simultaneously generate 4.4.9

$$\begin{aligned} V_{-1}(t) = \int_0^t \Big[& (I_{a\ new} * I_{fmg\ new} - I_{fm\ new} * I_{a\ old})^2 + \\ & (I_{fmg\ new} * N_{m\ new} - N_{mg\ old} * I_{fm\ new})^2 + \\ & (I_{ag\ old} * N_{m\ new} - N_{mg\ old} * I_{a\ new})^2 \Big] d\sigma \end{aligned} \qquad 4.4.9$$

6) The Gateaux differential is now computed as

$$D_G V(I_{fmg\ old}, I_{fm\ perturbed}) = (V_{+1} - V_{-1}) / 0.01 \quad .. \qquad 4.4.10$$

7) The value of $D_G V(I_{fmg\ old}, I_{fm\ perturbed})$ is stored and steps 1 to 7 are followed for I_{ag} and N_{mg}.

8) The trajectories are manipulated next by the rule

$$\left.\begin{aligned} I^{k+1}_{fmg} &= I^k_{fmg} - 0.0001 * D_G V(I^k_{fmg}, I^k_{fm\ perturbed}) / (1 + .1 * k) \\ I^{k+1}_{ag} &= I^k_{ag} - 0.0001 * D_G V(I^k_{ag}, I^k_{a\ perturbed}) / (1 + .1 * k) \\ N^{k+1}_{mg} &= N^k_{mg} - 0.0001 * D_G V(N^k_{mg}, N_{m\ perturbed}) / (1 + .1 * k) \end{aligned}\right\} 4.4.11^*$$

9) The steps 2 to 9 are repeated till the inequality 4.4.12 is satisfied.

For each n

$$|I_{fmg} - I_{fm}| + |I_{ag} - I_a| + |N_{mg} - N_m| \leq 0.0001 \qquad 4.4.12$$

---

* The multiplier 0.0001 limits the largest variation in trajectory to about 5%. The factor (1 + .1 * k) makes the step size a function of iteration number.

The inequality 4.4.12 corresponds to the condition that Gateaux differential of the function is minimum and function V has achieved the lower bound.

4.5 Computer Results

The Table 4.1 summerises all the results detailed in the Figures 4.5 through 4.7.

The main observation from theser esults is that higher loading of the main drive would be possible with the given specifications.

The Figures 4.8 through 4.10 demonstrate how the trajectories get manipulated when the algorithm described in Chapter III is used.

Figure 4.8a : shows the manipulation on the solution of the speed trajectory.

Figure 4.8b : shows the manipulation on the guess of the solution of the speed trajectory.

Figure 4.9a : shows the manipulation on the solution of the motor field current.

Figure 4.9b : shows the manipulation on the guess of the solution of motor field current.

Figure 4.10a : shows the manipulation on the solution of armature current.

Figure 4.10b : shows the manipulation on the guess of the solution of armature current.

Summary of the computer results.

Mode of operation	Speed		Field voltage		Field current		Terminal voltage		Armature current (max.)	Time of recovery
	Maximum error	Final error	Peak value	Final value	Peak value	Final value	Max./Min. value	Final value		
Generator field control mode										
zero to base	less than 0.03 p.u.	less than 0.01 p.u.	10.03 p.u.	1.0 p.u.	1.08 p.u.	1.02 p.u.	1.05 p. p.u.	1.0 p.u.	0.3 p.u.	
Loading at top speed	0.05 p.u.	less than 0.01 p.u.	9.8 p.u.	1.0 p.u.	1.08 p.u.	1.0 p.u.	Approx. 1.00 p.u.	Approx. 1.00p.u.	0.3 p.u.	Less than 0.5 sec.
Load throwoff at top speed	0.05 p.u.	0.0 p.u.	1.0 p.u.	1.0 p.u.	0.95 p.u.	1.0 p.u.	Approx. 1.00p.u.	Approx. 1.00p.u.	(Regenerative) 0.05p.u.	Less than 0.5 sec.
Motor field control mode										
base to top	0.06 p.u.	less than 0.02 p.u.	-10.5 p.u.	0.5 p.u.	1.33 p.u.	0.5 p.u.	Max. 1.05p.u. Min. 0.75p.u.	1.0 p.u.	0.4 p.u.	

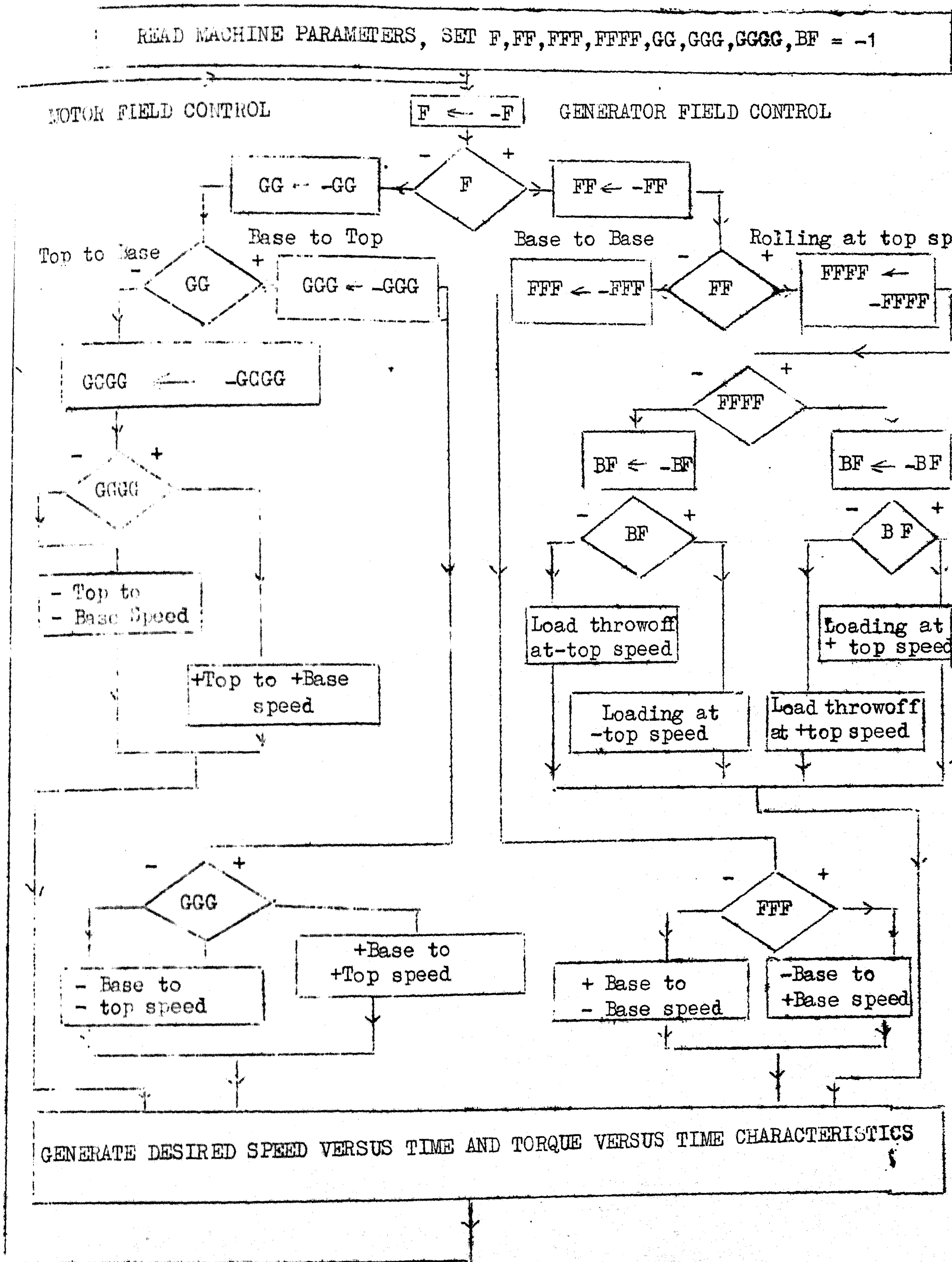


Figure 4-1 The rolling schedule.

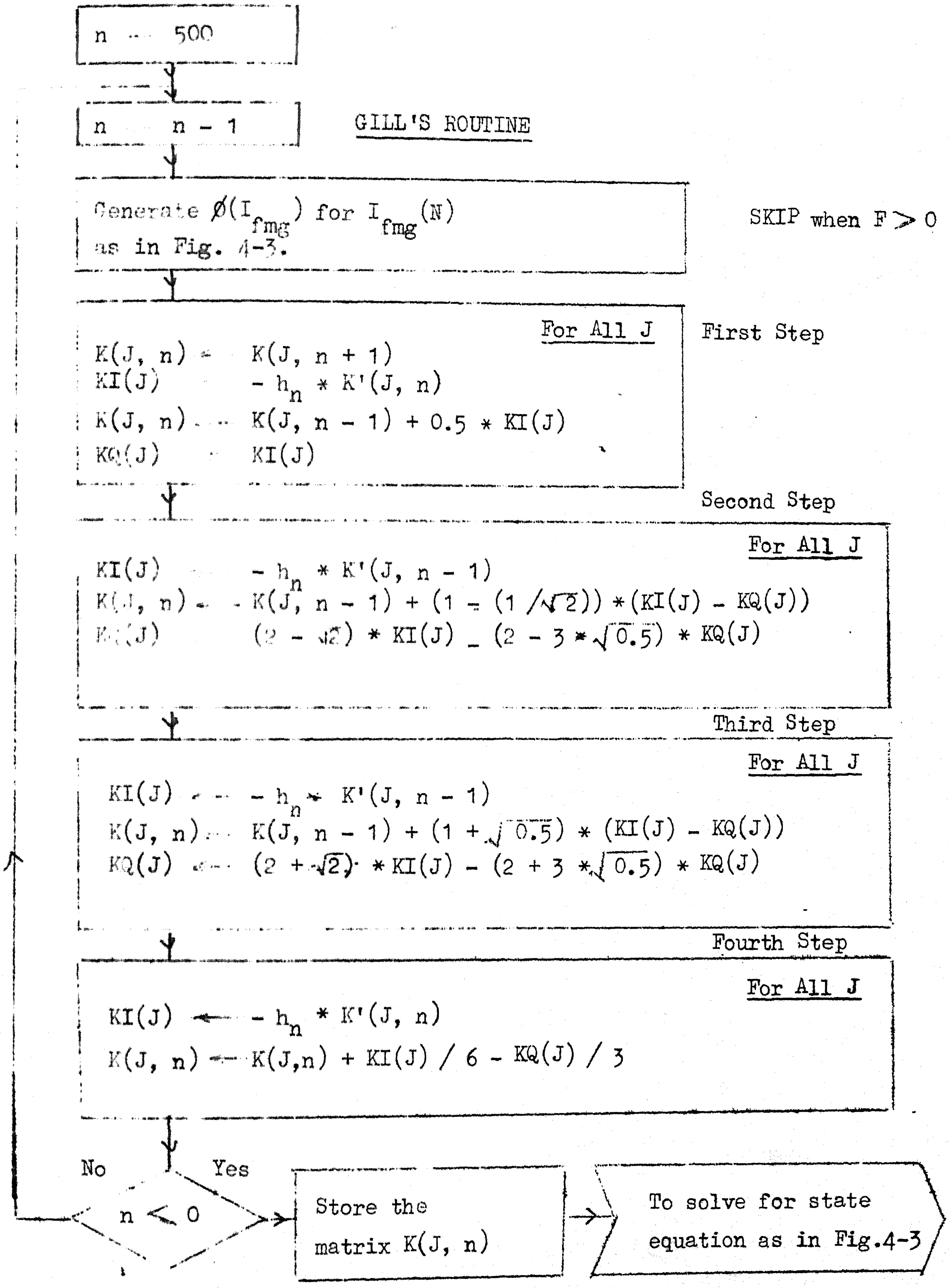


Figure 4-2

Solution of Auxiliary time varying differential equations.

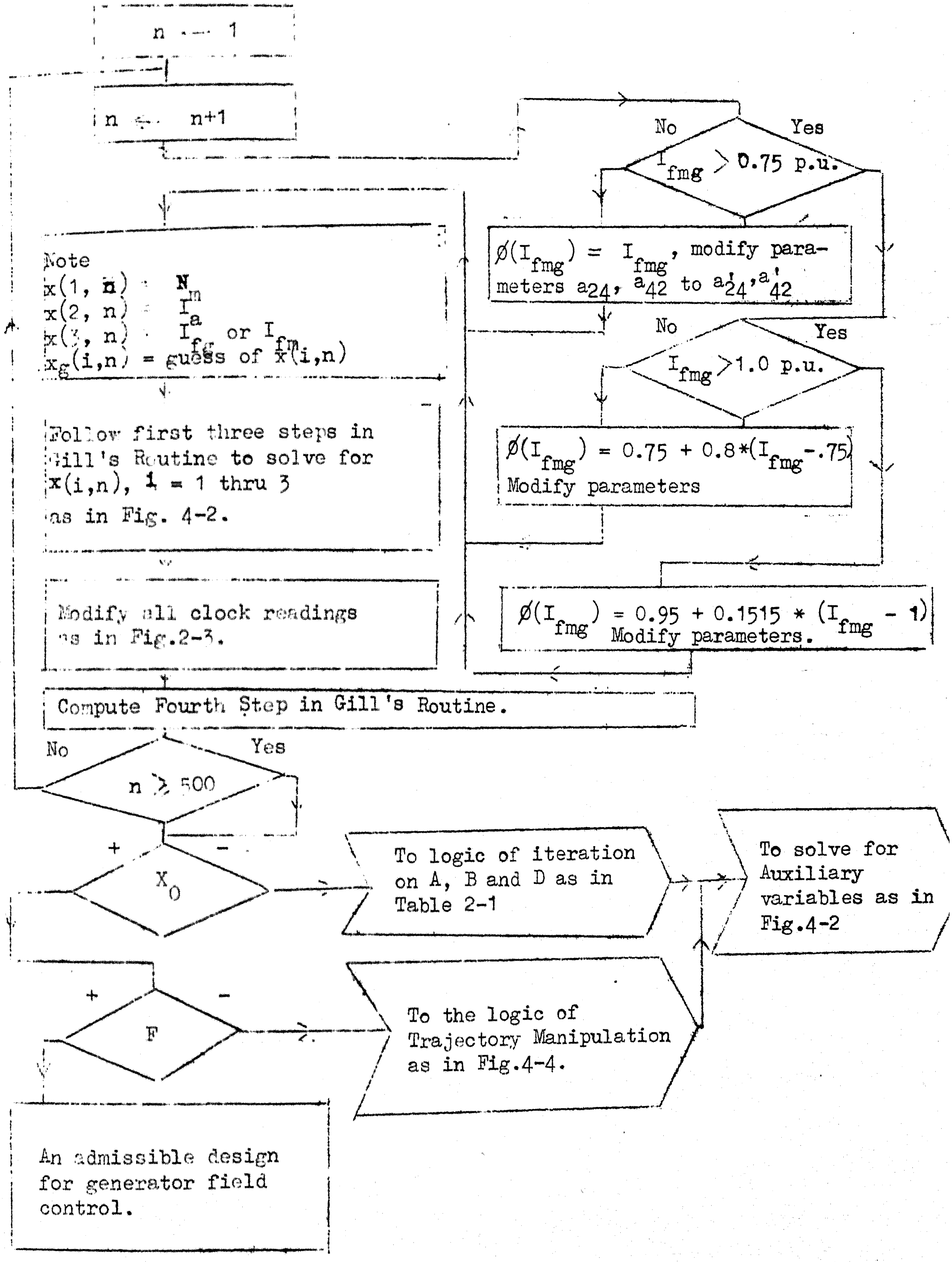
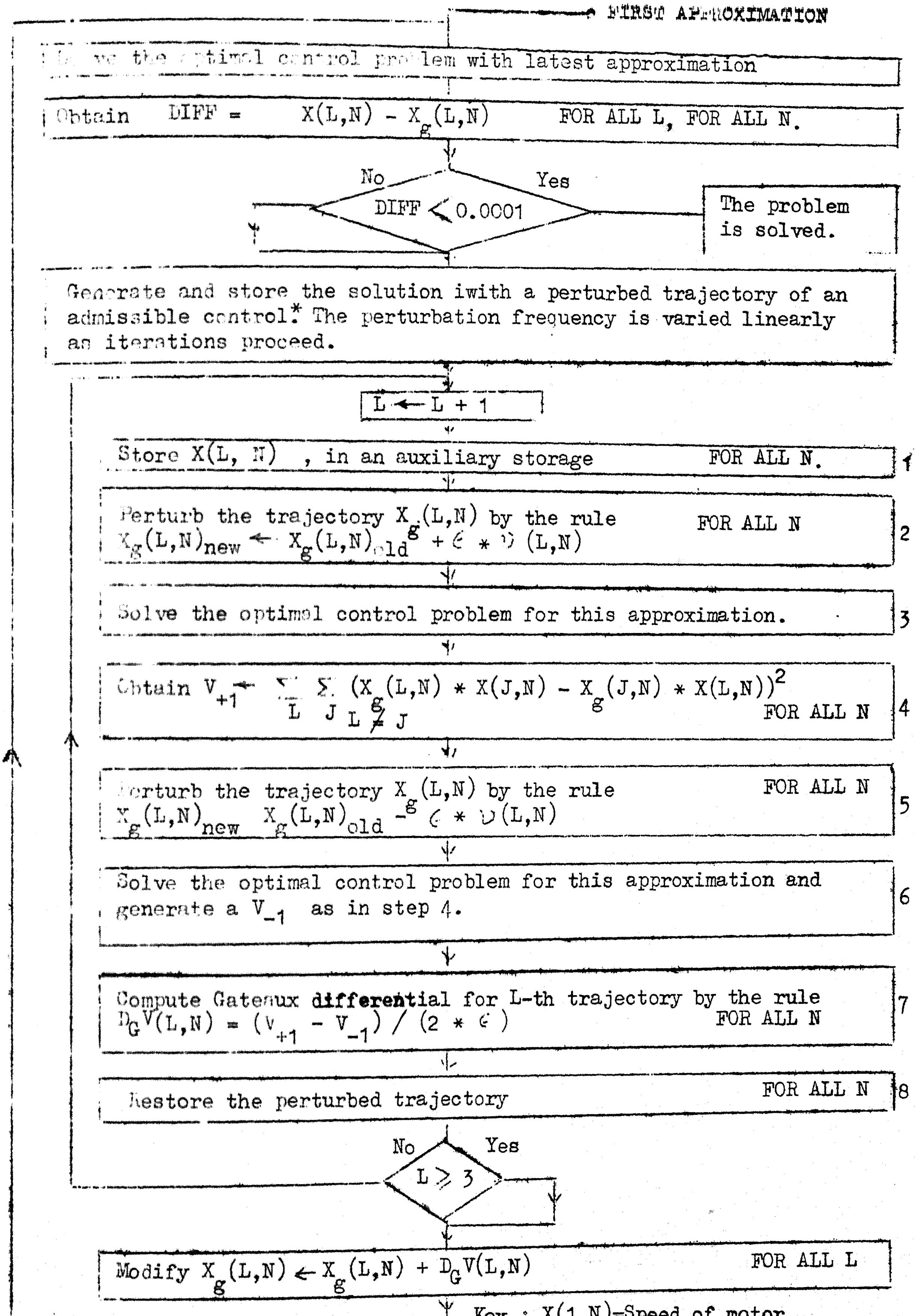


Figure 4-3

Solution of State Equations.



Key : $X(1,N)$=Speed of motor
$X(2,N)$=Armature current
$X(3,N)$=Motor field current
$X_g(L,N)$=Approximation to $X(L,N)$

Figure 4-4
Flow diagram for trajectory manipulation.
*Corresponding to analog computer solution, with sinusoidal perturbation.

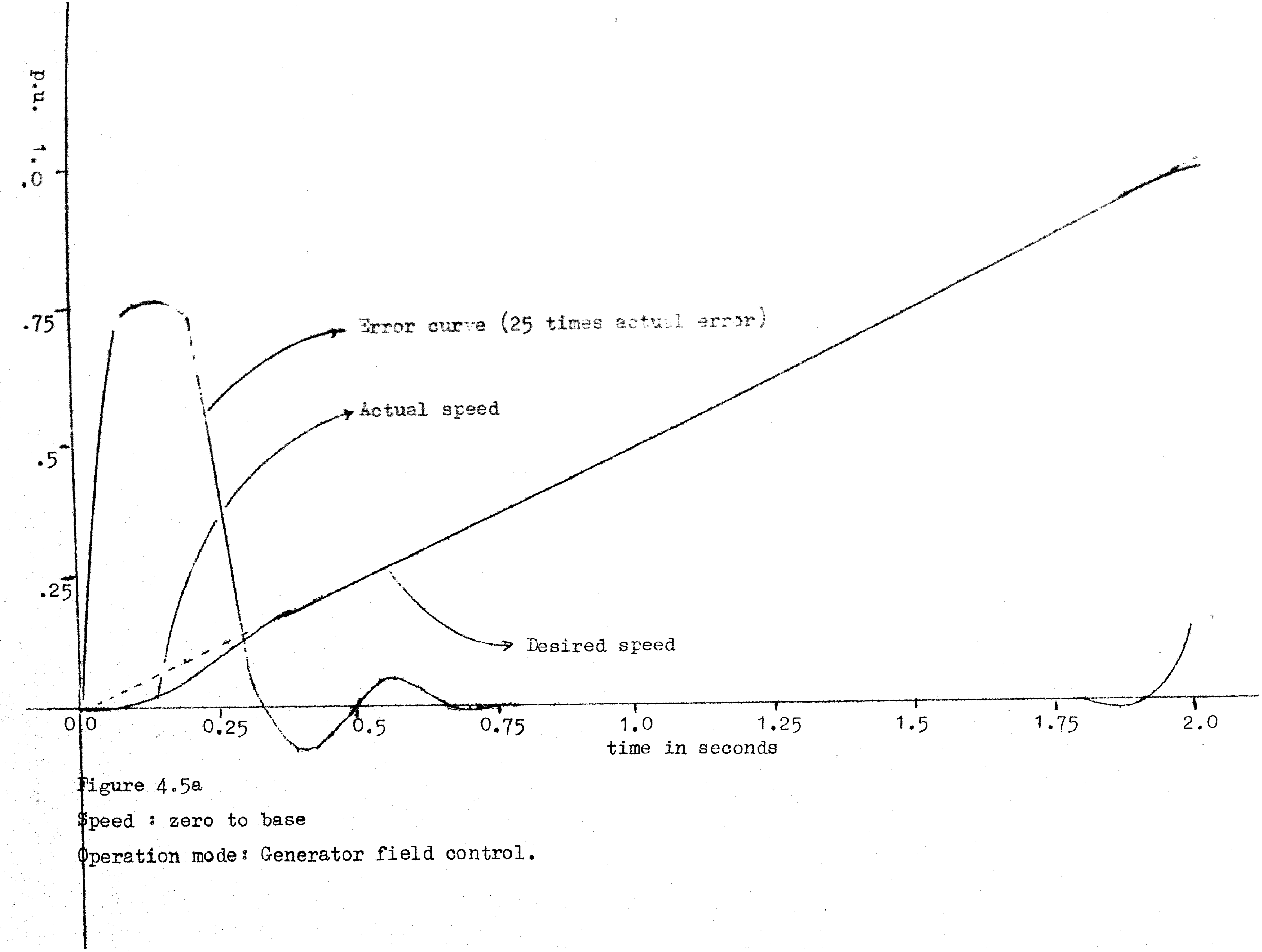


Figure 4.5a

Speed : zero to base

Operation mode: Generator field control.

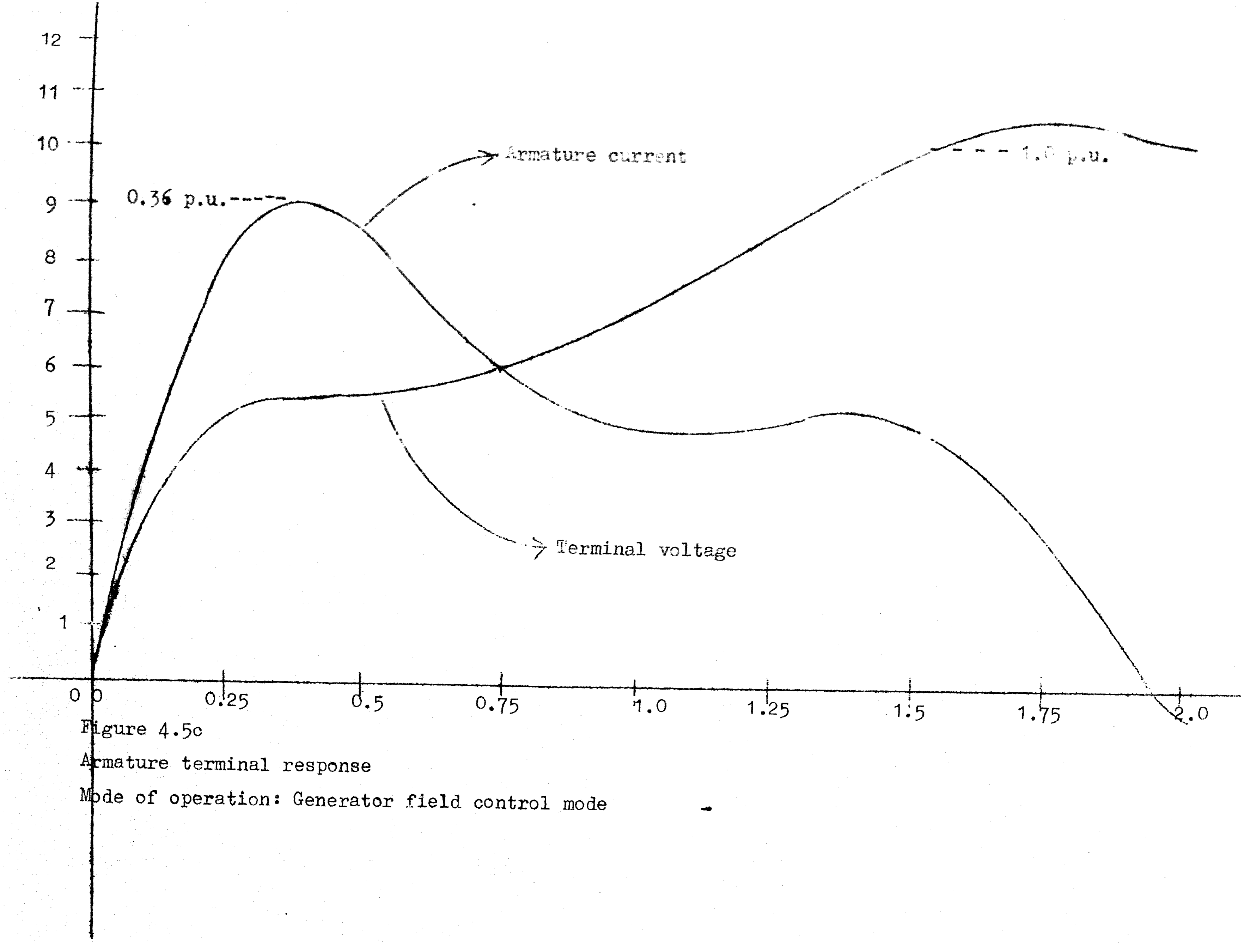


Figure 4.5c

Armature terminal response

Mode of operation: Generator field control mode

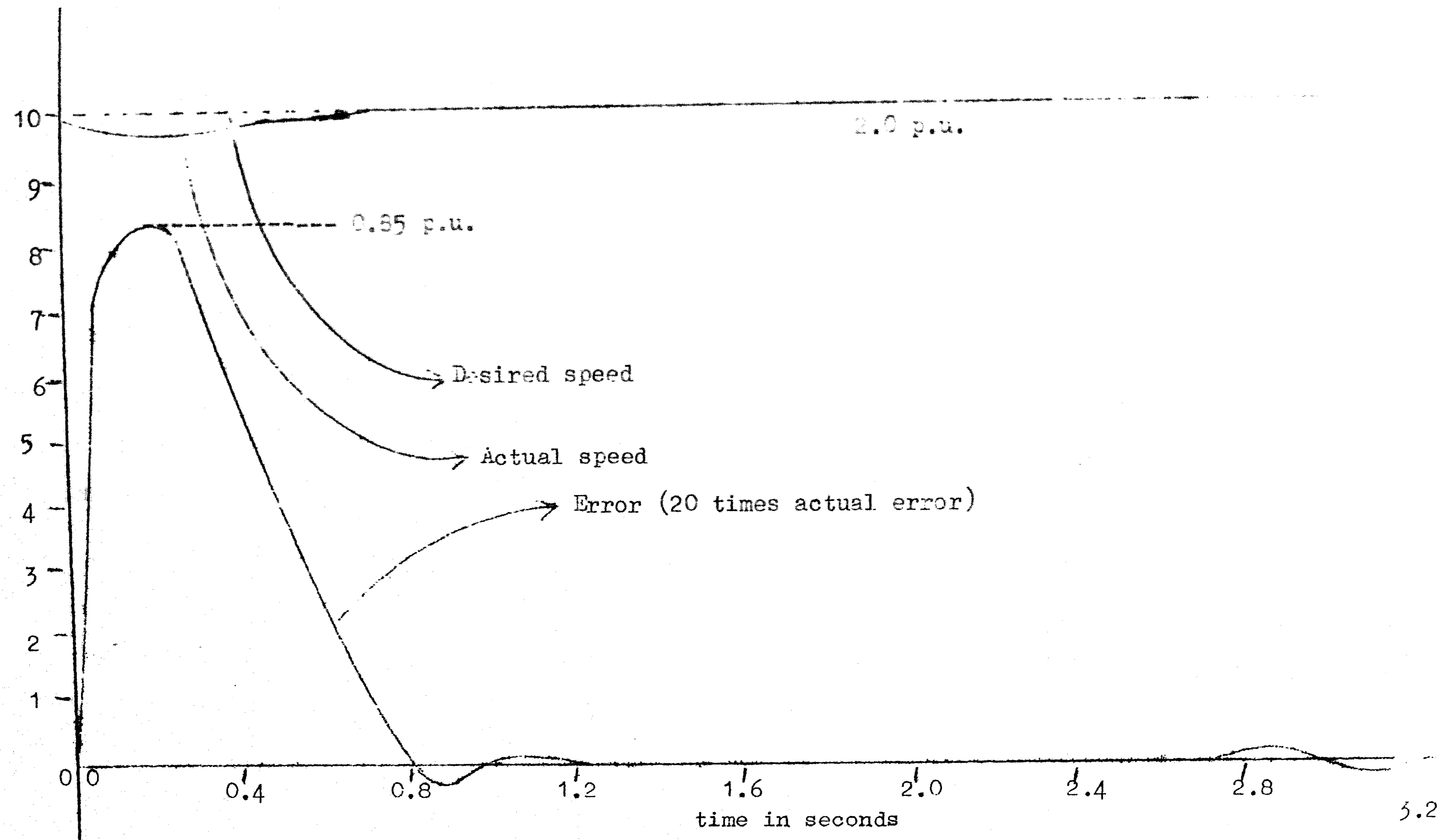


Figure 4.6a

Speed - top speed of operation

Mode of control - Generator field voltage.

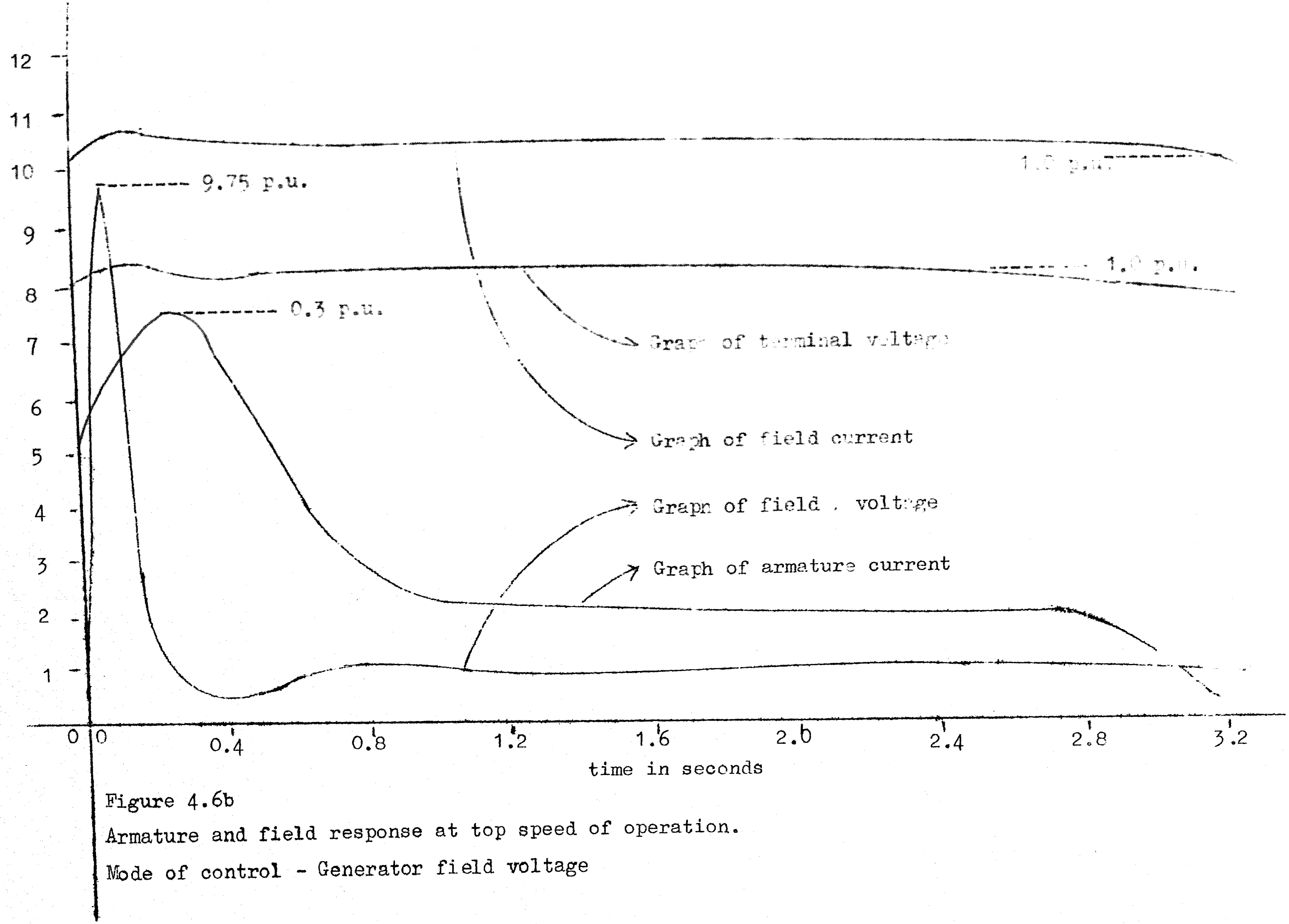


Figure 4.6b

Armature and field response at top speed of operation.

Mode of control - Generator field voltage

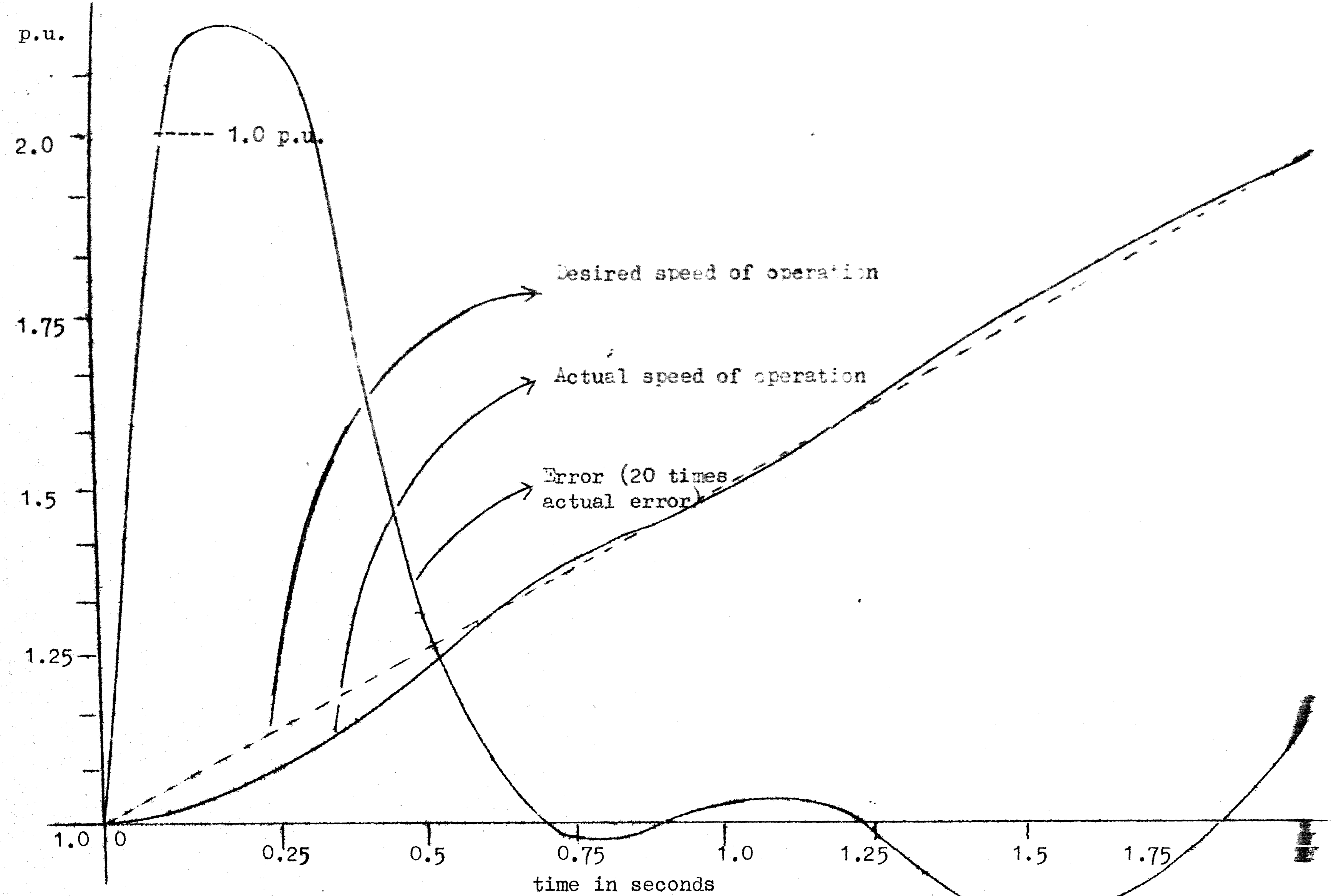


Figure 4.7a

Speed - Base to top speed

Mode of control - Motor field voltage control

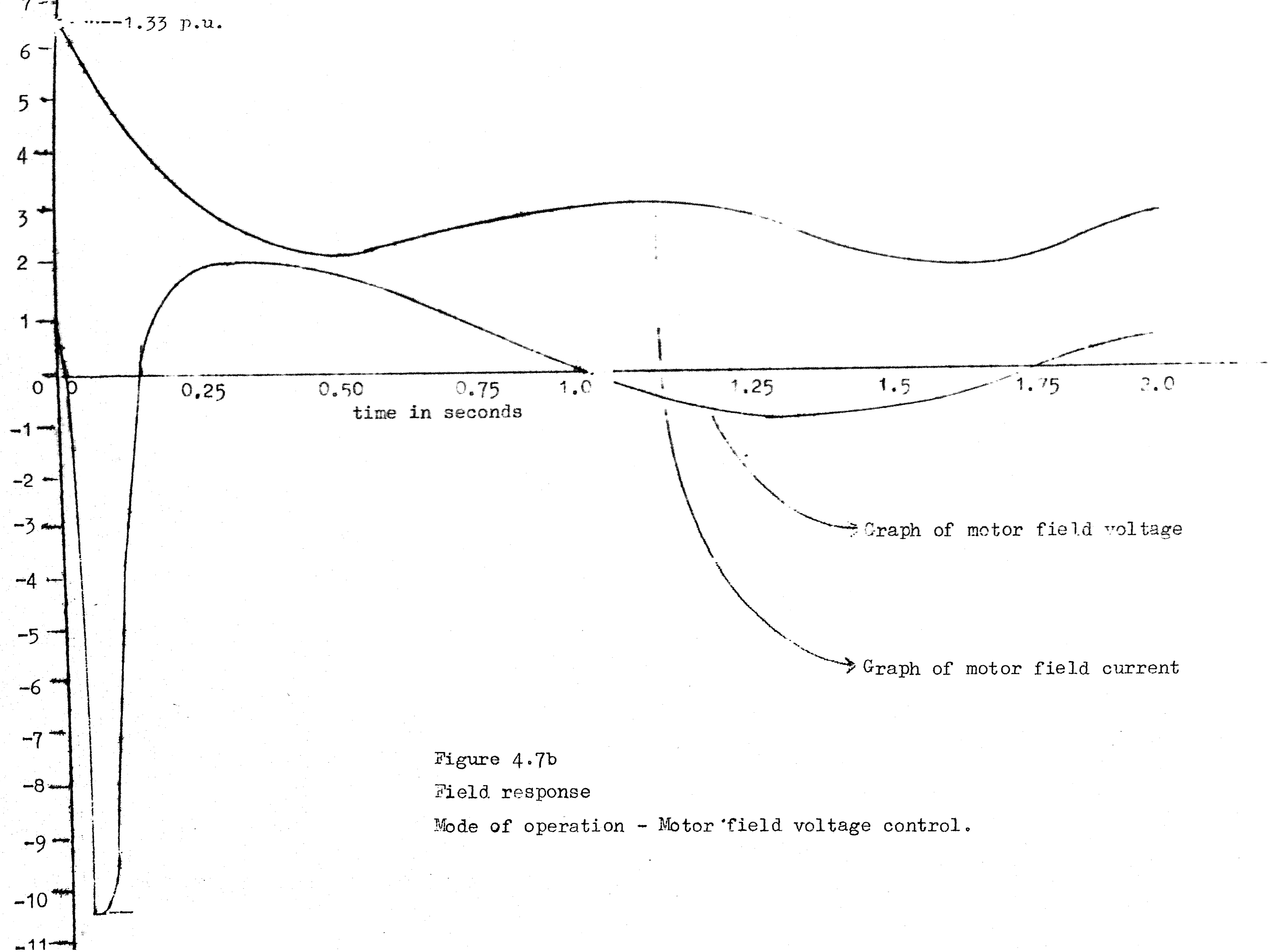


Figure 4.7b

Field response

Mode of operation - Motor field voltage control.

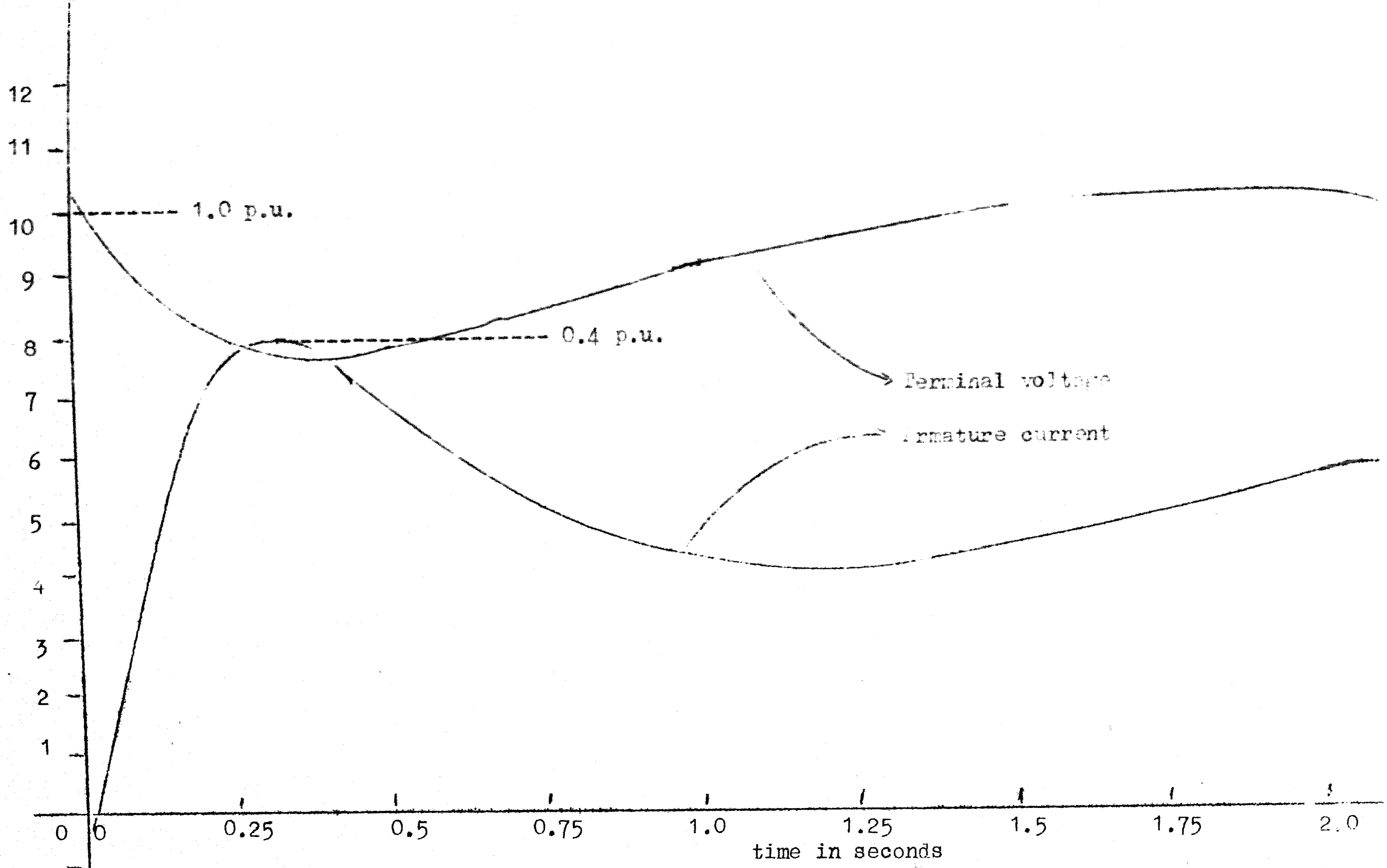


Figure 4.7c

Armature terminal response

Mode of operation - Motor field voltage control.

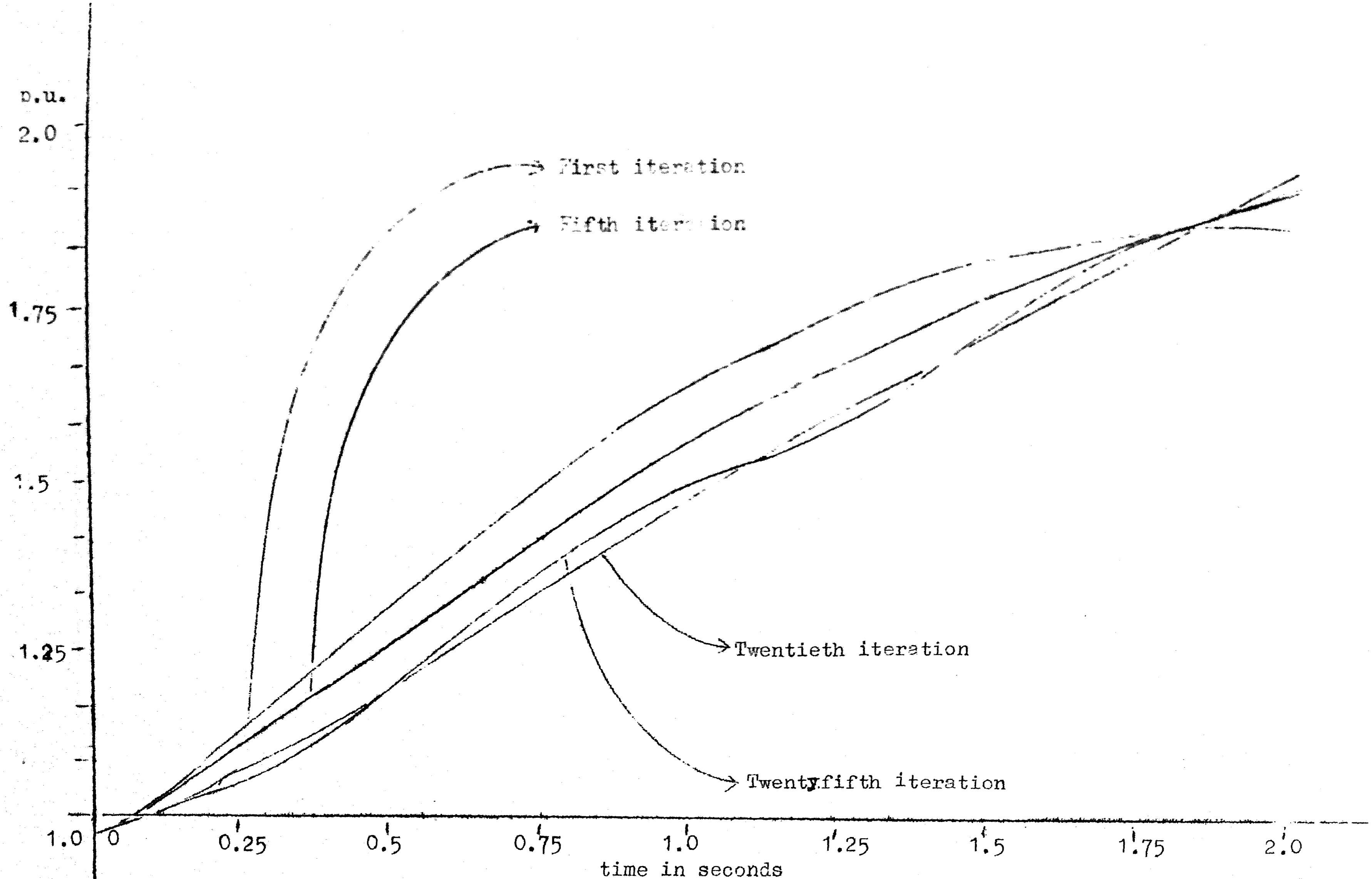


Figure 4.8a

Iterations on the speed trajectory.

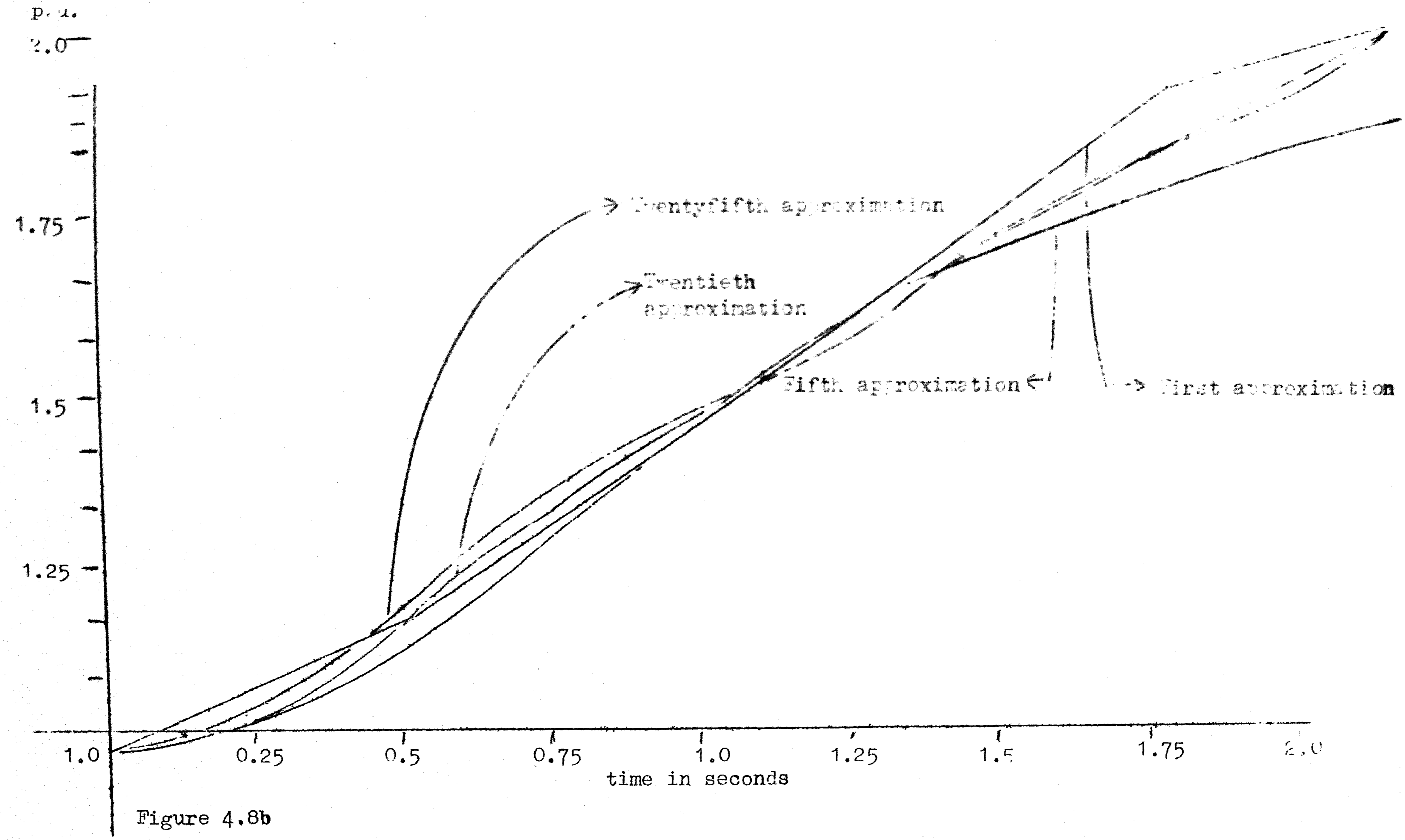


Figure 4.8b

Iterations on the approximation to the speed trajectory.

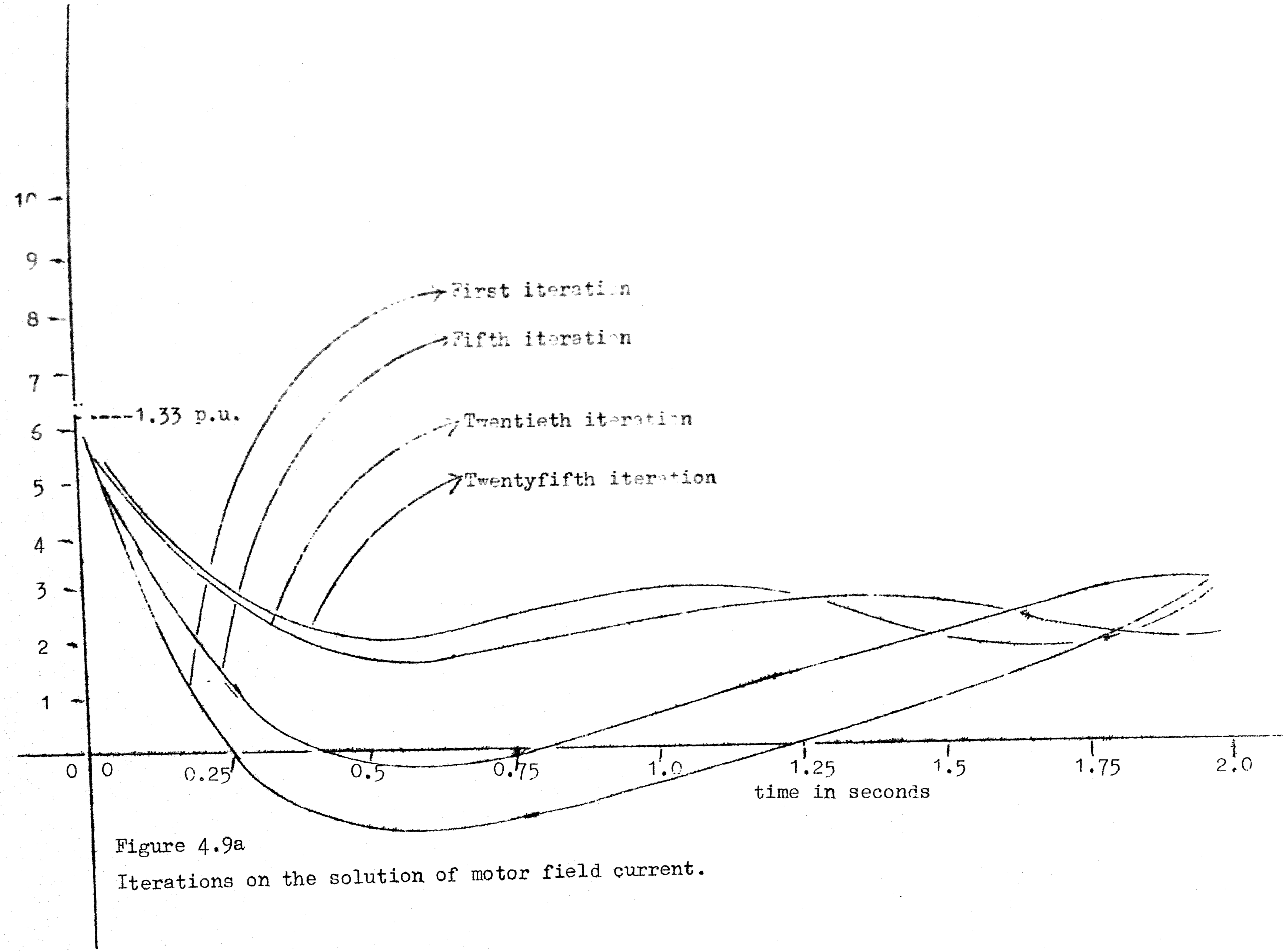


Figure 4.9a
Iterations on the solution of motor field current.

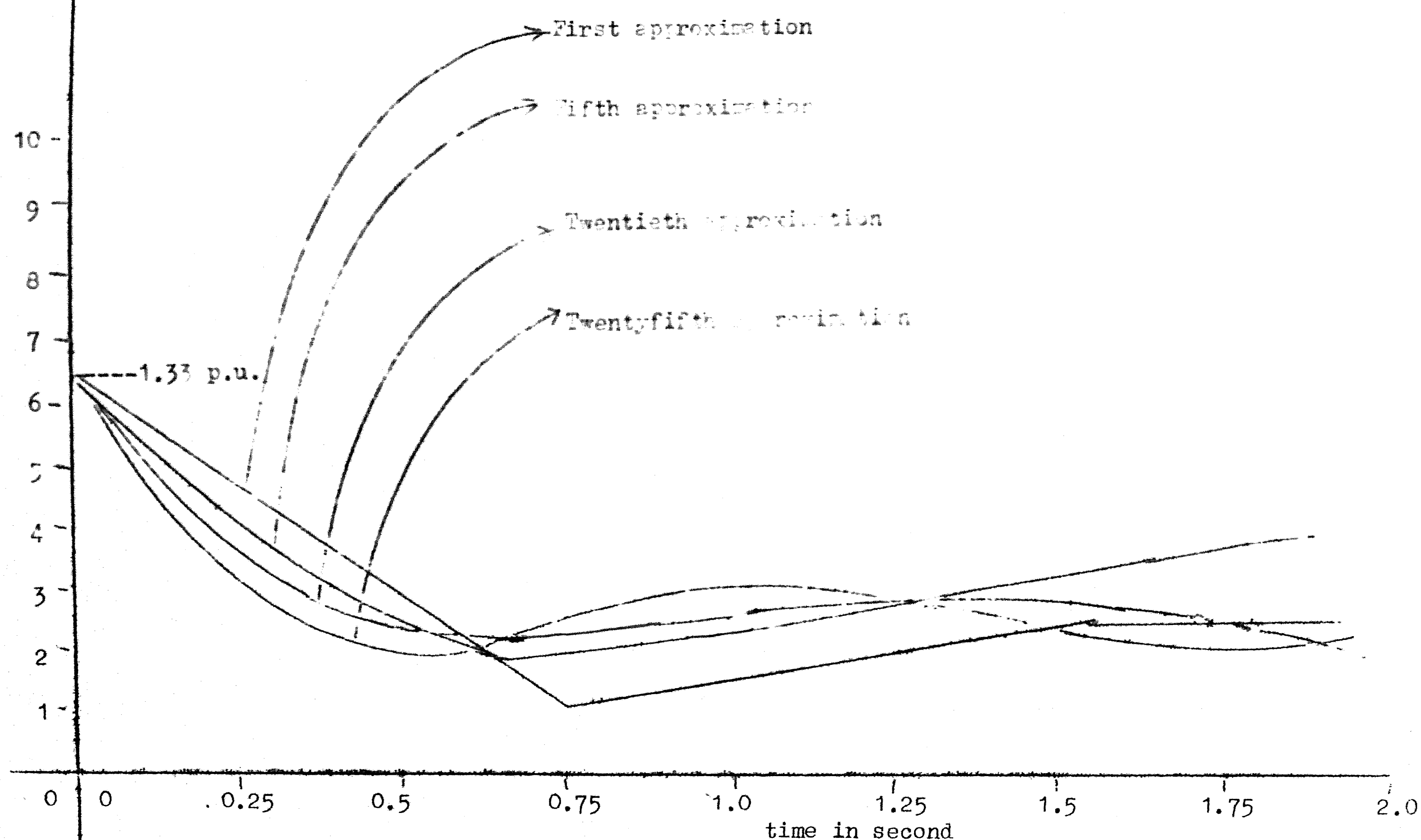


Figure 4.9b

Iteration on the approximations to the solution of motor field current.

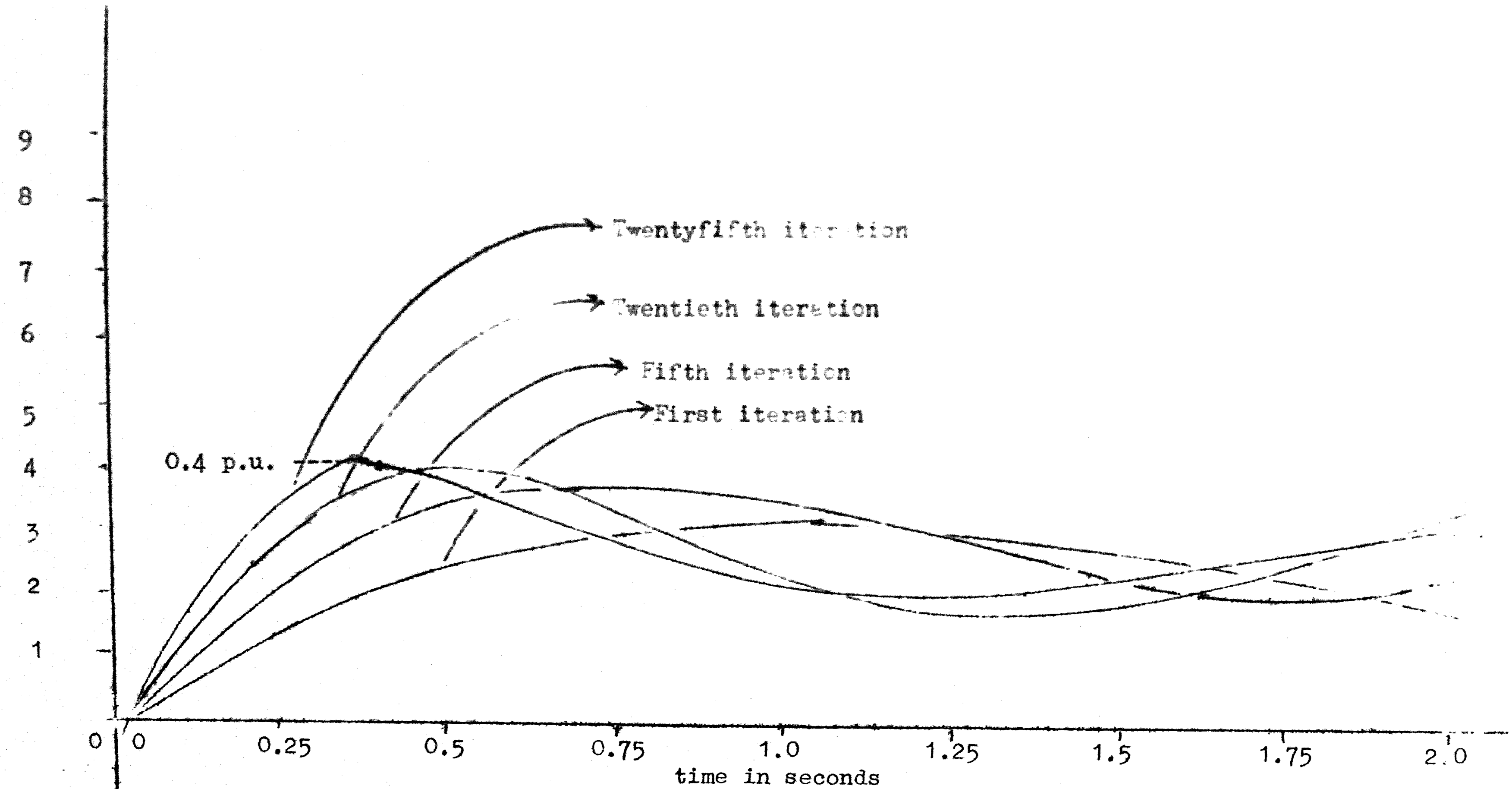


Figure 4.10a

Iteration on the solution of armature current.

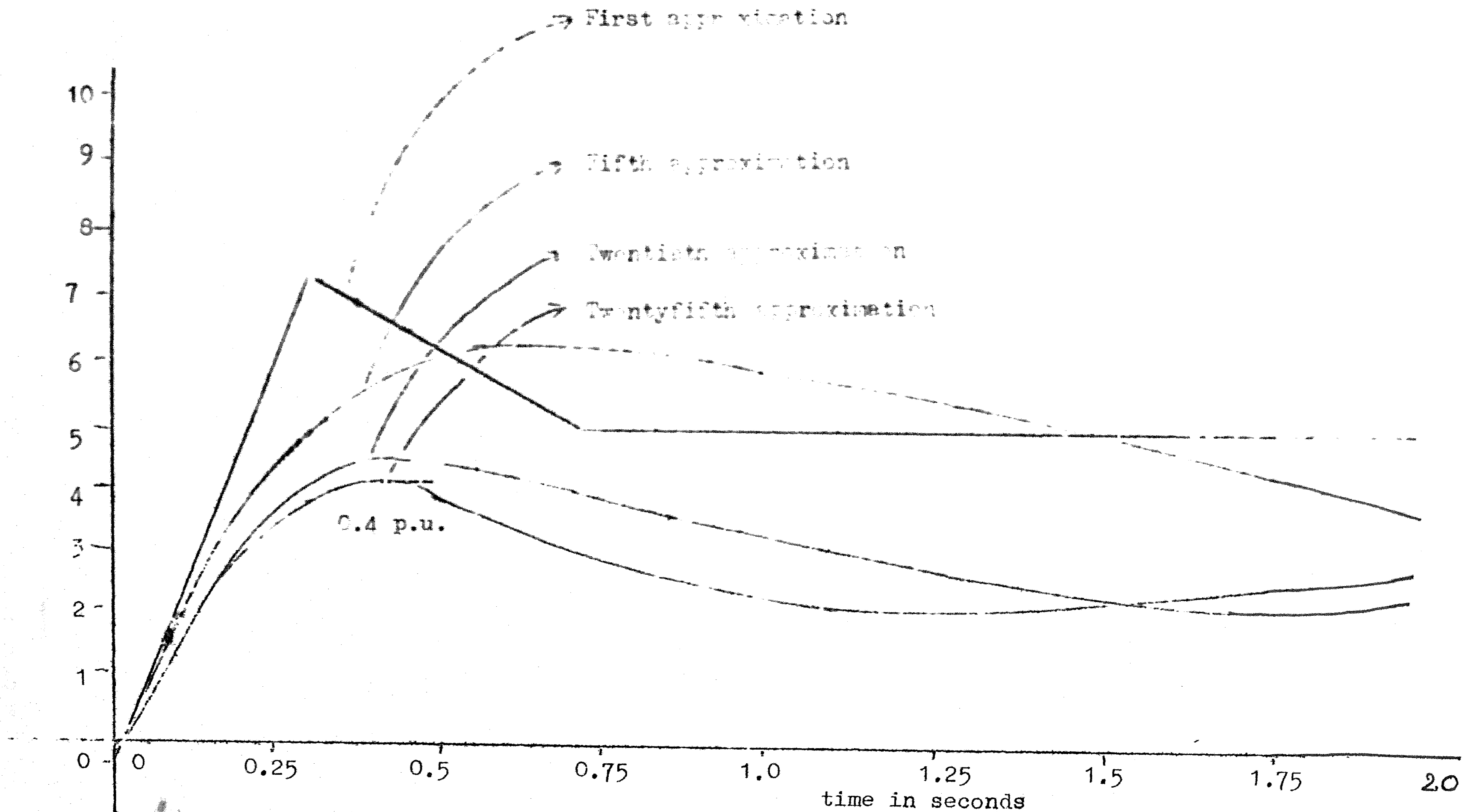


Figure 4.10b

Iteration on the approximations to the solution of armature current.

4.6 On-line Execution of Program[34,35,36,37,38]

There are two basic reasons why computer aided operations are recommended in practice :

1) The execution of control logic requires initial storage of appropriate initial conditions of the auxiliary variables. During the motor field voltage control mode,storage of the guesses of the state variable trajectories will also be required.

2) The torque versus time characteristics will vary as the rolling proceeds and therefore during the handling time*[34] on the work piece the torque versus time characteristics will have to be computed and predicted for the next pass. Also during this time the initial conditions of the auxiliary time variables will have to be computed.

It may be remarked here that the modes of operation of the mill main drive control system has following fixed patterns :

a) during the top speed : Generator field voltage control

b) between the top and base speed : Motor field voltage control

c) between ± base speed : Generator field voltage control.

These patterns follow with a certain regularity because of the nature of the rolling operations involved. The regularity of the patterns makes the use of computer aided operation attractive

---

* Time required to position the work piece or some auxiliary controller.

because of the possibility of introducing 'learning capabilities' as a consequence of data logging facilities available with modern process control computers like Prodec 580[8] or Tosbac-7000. This is explained in details in the Figures 4-11 through 4-13. In Figures 4-11 through 4-13 it is assumed that a single central processor will be required to achieve execution of number of programs (simultaneously residing in memory) in such a way that none of the programs need be completed before another is started or continued. Such a facility is generally termed as multi programming facility. This is desireable because though operations are scheduled by the main program or an executive program, the computer should be ready to receive and transmit relevent information within acceptable response time[35].

Figure 4-13 details the use of data logging in introducing the learning capability in the overall system. It also explains that in general, the dynamic program modification,if needed, ean be carried out.

In brief the computer ought to be a real-time* system.

---

* Real-time (ideally) is an extreme case of on lineness where the response time of the computer system shall be zero.

because of the possibility of introducing 'learning capabilities' as a consequence of data logging facilities available with modern process control computers like Prodec 580[8] or Tosbac-7000. This is explained in details in the Figures 4-11 through 4-13. In Figures 4-11 through 4-13 it is assumed that a single central processor will be required to achieve execution of number of programs (simultaneously residing in memory) in such a way that none of the programs need be completed before another is started or continued. Such a facility is generally termed as multi programming facility. This is desireable because though operations are scheduled by the main program or an executive program, the computer should be ready to receive and transmit relevent information within acceptable response time[35].

Figure 4-13 details the use of data logging in introducing the learning capability in the overall system. It also explains that in general, the dynamic program modification,if needed, can be carried out.

In brief the computer ought to be a real-time* system.

---

* Real-time (ideally) is an extreme case of on lineness where the response time of the computer system shall be zero.

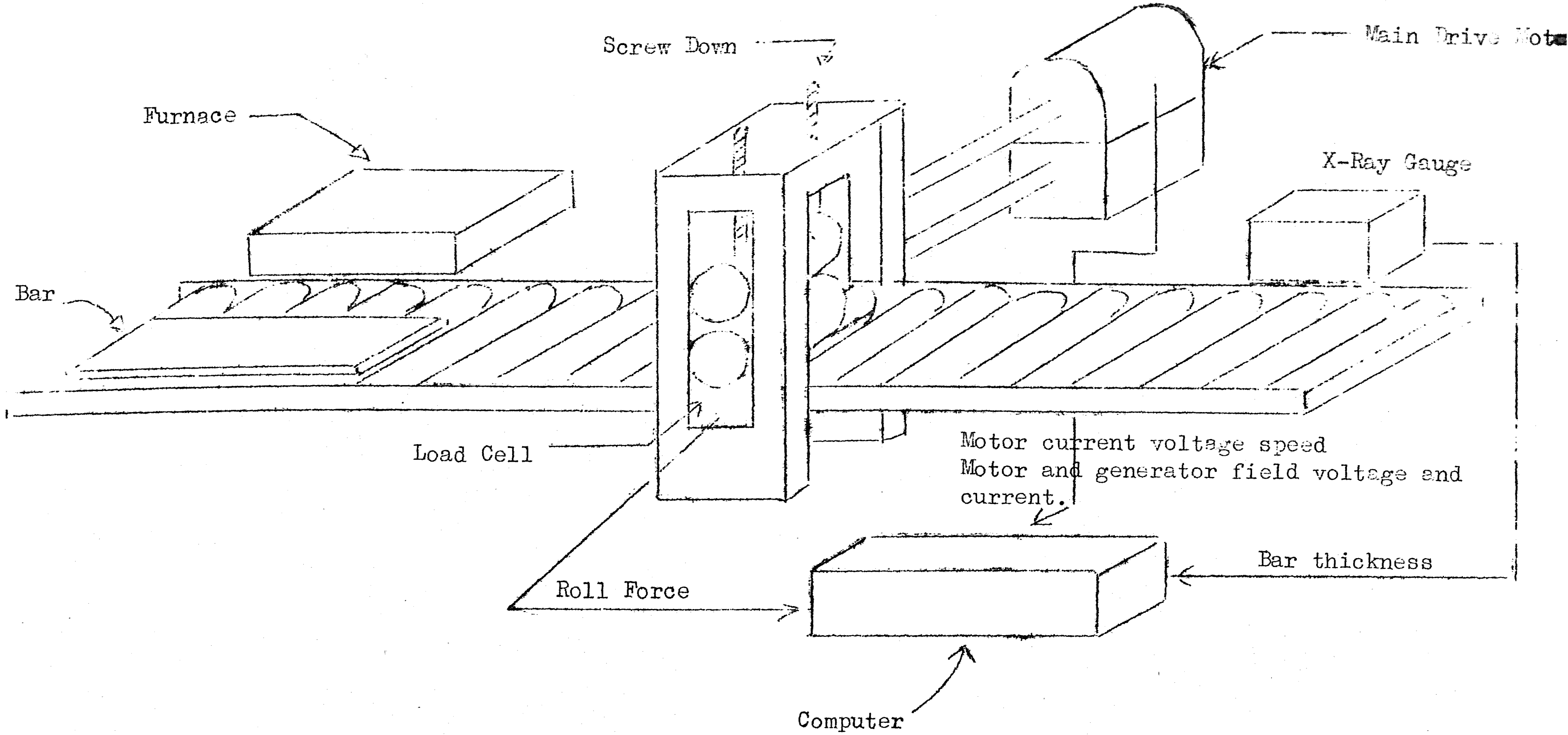


Figure 4-11

General arrangement of computer aided bar mill.

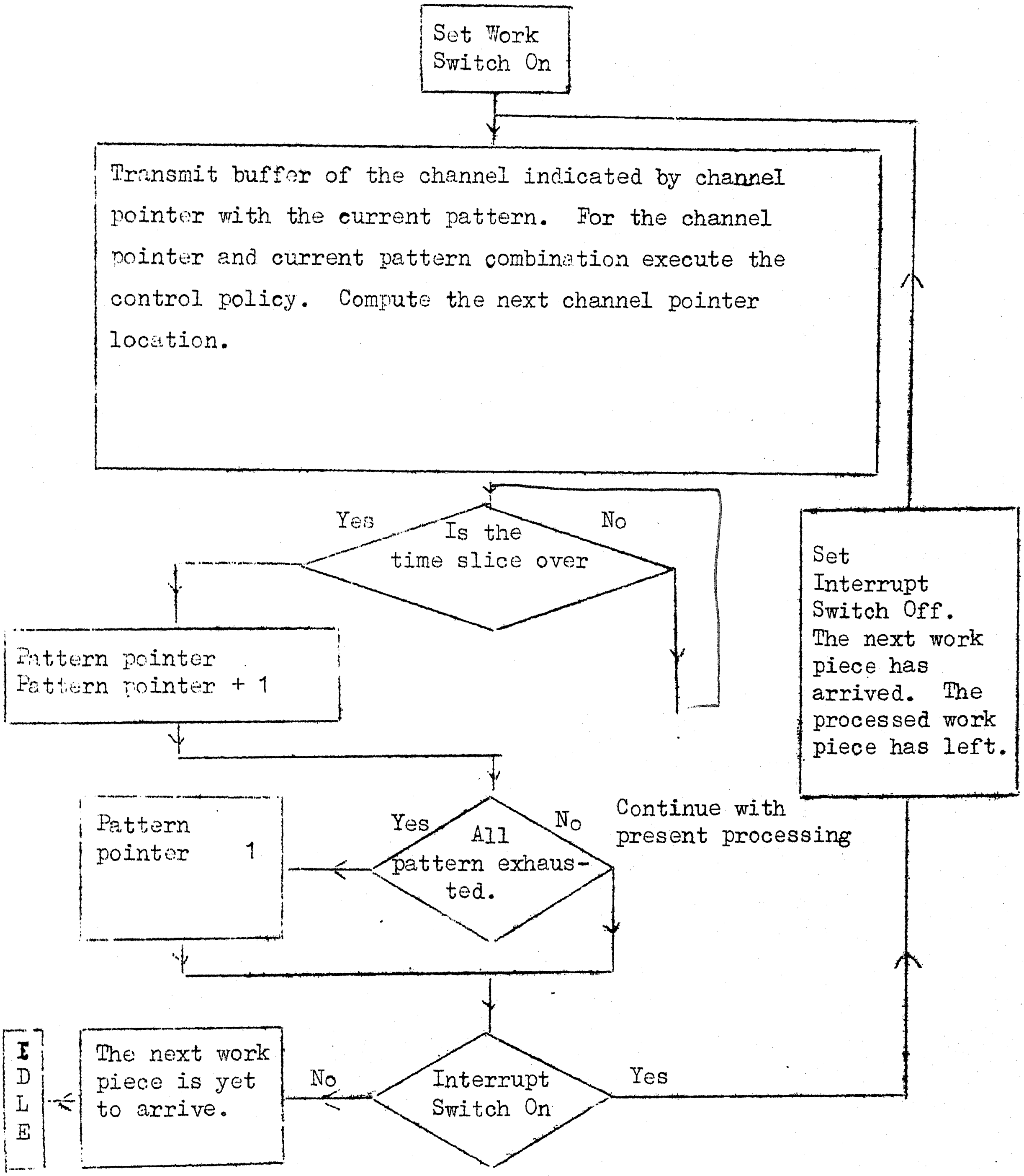
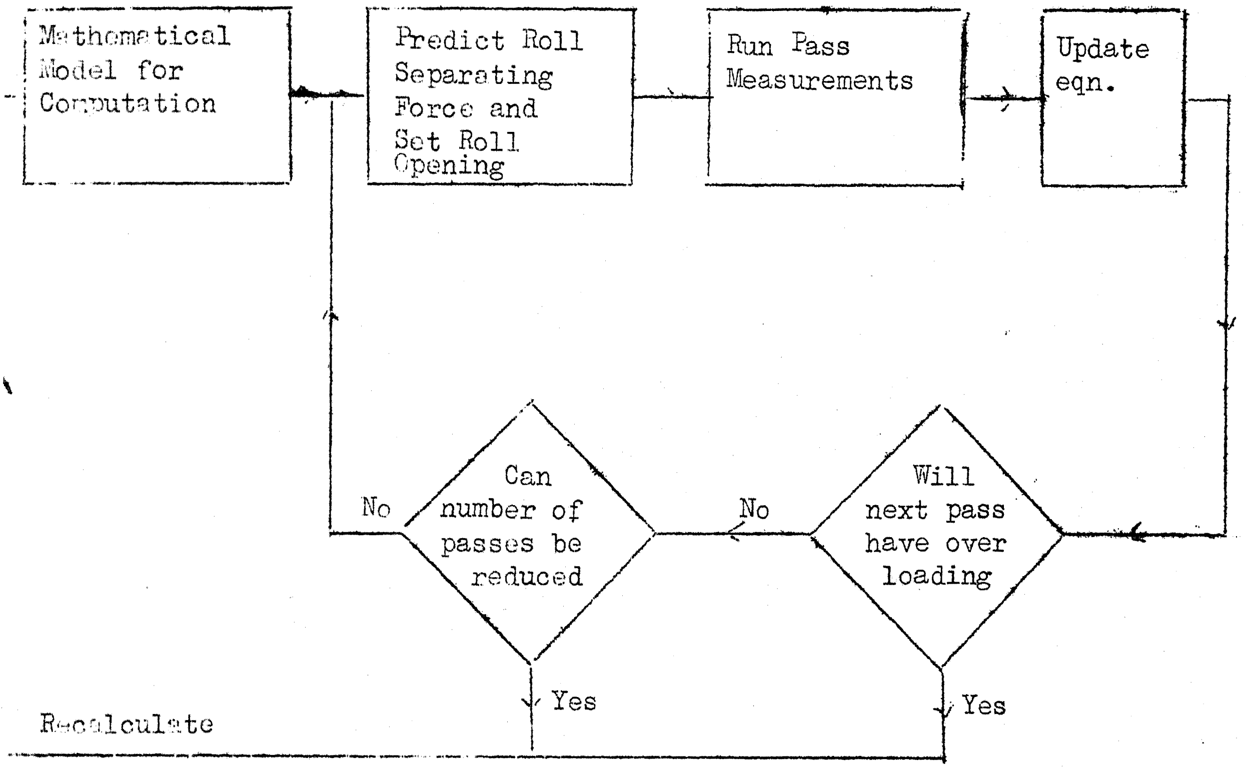


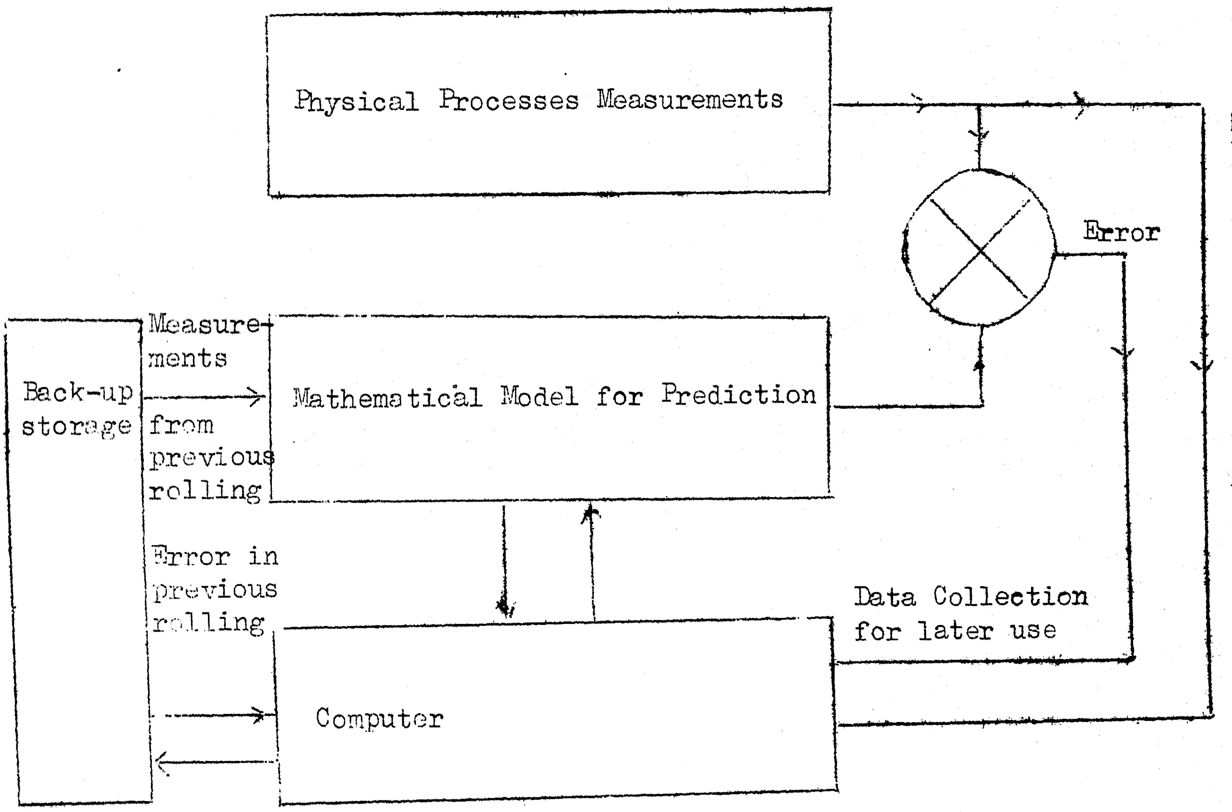
Figure 4-12

Flow diagram of the operation of the computer system.



Updating of Mathematical model.

Figure 4-13: Data logging and program modification.
Error Corrections in Successive Passes



CHAPTER - V

CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH

In this chapter it is proposed to cite the main results in the thesis and list some problems which may prove interesting for further investigation.

5.1 Main Conclusions

The main conclusion from this investigation is that,with the experience accumulated on nature of operation of rolling mill main drives, it should,in general, be possible to use process control computer to schedule and optimally regulate the speed at which the rolling is expected to take place. Since the control logic is based on a performance index which has associated weightages for the speed, terminal voltage (essentially as a function of state variables and machine parameters), the protection system suggested by clock records of the duration of abnormal condition, is not only feasible but is reliable also. An observation of computer results reveal following :

1) Speed over-shoots are almost completely eliminated.

2) There is no regenerative current during field weakening.

3) Variations in terminal voltage are larger than those usually observed during such operations.

One does not have to get greatly concerned about the terminal voltage variations as regulation of terminal voltage is not an

important consideration here. This is so because of isolation of armature terminals from the mains. This,of-course, will be a major problem when one machine complex is used with heavy rectifying equipment connected to mains.

This thesis is a significant contribution to the design approach and technique used to solve optimal control problem associated with systems whose state equations contain single-valued functional non-linearity and/or product type of non-linearity. The trajectory manipulation technique suggested in the thesis is general enough to be adapted for trajectory optimisation problems also.

5.2 Directions for further Research

The following is the list of some interesting extensions of the problem discussed in the thesis :

1) Feasibility studies to examine compatibility of present day process control computers for use in speed regulation as suggested in the thesis.

2) It will be worth-while to examine how the performance of the mill main drive compares (with the present results), with the condition when one is permitted to have discontinuous forcing functions.

3) An immediate extension of this thesis would be to examine the class of problems which can be solved by employing the trajectory manipulation as suggested. A computational comparison of this method with those available in literature will also be revealing in many respects.

LIST OF REFERENCES

1. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V. and Mischenko, E.F.: "The Mathematical Theory of Optimal Processes", Interscience Publishers Inc., New York, 1962 (Book).

2. Bellman, R.: "Dynamic Programming", Princeton University Press, Princeton, N.J., 1957 (Book).

3. Brown, R.R.: "A Generalised Procedure for the Design of Optimum Systems" - Part-I, Trans. A.I.E.E., Vol.78, Pt.-1, 1959, pp. 285-289.

4. Stringer, L.F., Schonholzer, E.T.: "Thyristor Drive Systems", Journal of Iron and Steel Engineer, March 1967.

5. Hansan, A., Wilkerson, A.W.: "Automatic Speed regulation of D.C. Motors Using Combined Armature Control and Motor Field Control", Trans. A.I.E.E., (App. and Ind.), May 1961, pp. 59-64.

6. Plaskett, E.H.: "Closed-loop Regulating Systems - the Modular Approach", A.E.I. Engineering, Sept./Oct. 1966.

7. Bhatt, P.C.P.: "Modelling and Simulation of a Bar Mill", being considered for publication, Institut belge de re'gulation et d' Automatisc, Bruxells, Belgium.

8. "The Completely Automated Hot Strip-mill", Industrial Systems Division, Westinghouse Electric Corporation, Pittsburgh 35, Pennsylvania, (Manual).

9. Okuda, N., Anbe, Y., Ushiku, O.: "Computerised Adaptive Control of a Hot-Strip Mill", Electrical Engineering in Japan, Vol.88, No.7, 1968, pp. 60-68.

10. Merriam, C.W.III: "Optimisation Theory and the Design of Feed-back Control Systems", McGraw-Hill Book Co. New York, 1964, (Book).

11. Larke, E.C.: "The Rolling of Metals - Theory and Experiment", Chapman and Hall, London, 1963, (Book).

12. U.S.Steel Corporation: "Making Shaping and Treating of Steel", 1965, (Hand Book).

13. Derusso, P.M., Roy, R.J., Close, C.M.: "State Variables for Engineers", John-Wiley & Sons Inc. New York, 1965, (Book).

14. Merriam, C.W.III: "Use of a Mathematical Error Criterion in the Design of Adaptive Control Systems", Trans. A.I.E.E. (App. and Ind.), Paper No. 59-1158, January 1960.

15. Ellert, F.J., Merriam, C.W.III, "Synthesis of Feed-back Controls using Optimisation Theory - An Example", Paper presented at J.A.C.C.,June 29, 1962.

16. Rekasius, Z.V., Hsia, T.C., "On an Inverse Problem in Optimal Control", Trans. I.E.E.E. (Automatic Control), Vol.AC-9, No.4, October, 1964, pp. 370-375.

17. Rajaraman, V., "Principles of Fortran Programming", Prentice Hall of India, 1969, (Book).

18. Sage, A.P.: "Optimum System Control", Prentice Hall Inc., New Jersey, 1968,(Book).

19. Gill, S.: "A Process for the Step-by-Step Integration of Differential Equations in an Automatic Digital Computing Machine", Proc. Cambridge Phil. Soc., Vol.47, Pt.-1, June 1950.

20. Bryson, A.E., Carrol, F.J., Mikami, K., Denham, W.: "Determination of the Lift or Drag Program That Minimises Re-entry Heating With Acceleration or Range Constraints using a Steepest Descent Computation Procedure", Proc. I.A.S., 29th Annual Meeting, New York, 1961.

21. Kopp, R.E., McGill, R., Moyer, H.G., Pinkham, G. in "Computing Methods in Optimisation Problems" (Balkrishnan, A.V., Neustadt, eds.), Academic Press, New York, 1964, pp.65-106.

22. Lukes, D.L.: "Steepest Descent and Maximum Principle Techniques of System Optimisation", U-RD6195, Minneapolis - Honeywell Regulator Company, Minneapolis, Minnesota, Dec. 1961.

23. Jazwinski, A.H.: "Steepest Descent Trajectory Optimisation with Inequality Constraints", Journal of Aero-space Sciences, Vol.29, Oct. 1962.

24. Merriam, C.W.III: "An Algorithm for the Iterative Solution of a Class of Two-Point-Boundary-Value Problems", Journal of SIAM (Control) Ser. A., Vol.2, No.1, 1964.

25. Mitter, S.K.: "Successive Approximation Methods for the Solution of Optimal Control Problem", Automatica, Vol.13, No.3/4, Jan. 1966, pp. 135-150.

26. McGill, R., Kenneth, P.: "Two Point Boundary Value Techniques", in Advances in Control Systems (Theory and Applications), ed.by Leondes, C.T., Vol.3, 1966.

27. Mitter, S.K., Lasdon, S.L., Waren, A.D.: "The Conjugate Gradient Method for Optimal Control Problems", Trans. I.E.E.E. (Automatic Control), Vol.AC-12, April 1967.

28. Fletcher, R. and Powell, M.: "Functional Minimisation by Conjugate Gradients", Computer Journal, Vol.6, 1964, pp. 163-168.

29. Luenberger, D.G.: "Optimisation by Vector Space Methods", John Wiley and Sons Inc., New York, 1968, (Book).

30. Ramaratnam, V.S., Sinha, V.P.: "Analog Computer Simulation Study of 28" Bar Mill", Computer Study C.S.-1, Applications Engineering Division, Heavy Electricals (India) Ltd. Bhopal.

31. White, D.C. and Woodson, H.H.: "Electromechanical Energy Conversion", John Wiley Inc. New York, 1959, (Book).

32. Roe, R.H.: "Network and Systems", Addison Wesley, 1966, (Book).

33. Gorn, S., Moore, R.: "Automatic Error Control - The Initial Value Problem in Ordinary Differential Equations", Ballistic Research Laboratory Report No.893, Aberdeen, Proving Ground, Md., 1954.

34. Bhatt, P.C.P. and Mishra, J.: "Computer Aided Rolling", EE Research Report, I.I.T. Kanpur, October 1969.

35. Aron, J.D.: "Real-Time Systems in Perspective", IBM Systems Journal, Vol.6,No.1, 1967, pp. 49-67.

36. Martin, J.: "Design of Real-Time Computer Systems", Prentice-Hall, 1967, (Book).

37. Cooper, W. and Heckathorne, R.: "Hardware for On-Line" in The Transition to On-Line Computing (Problems and Solutions), ed. by Gruenberger, F., Thompson Book Company, 1967, pp.39-64.

38. Ushiku, O.: "Computer Controlled Hot-Strip Mill (The Computer Control System and a Mathematical Model)", Toshiba Review, July-September 1966, pp. 1-6.

APPENDIX - A

An Analog Computer Study of Plaskett's Scheme

The main purpose of this appendix is to demonstrate existance of a feasible solution for the problem stated in Chapter II. The appendix describes a particular investigation which was carried out to find compensation parameters for a speed regulating system identical to Plaskett's[6].

A.1 Drive Equipment and Performance Requirements

Same as described in Chapter II.

The schematic arrangement is detailed in Fig.A-1.

A.2 The Logic of Regulating System Design

Though this system has been designed primarily to regulate the speed, limitations on machine design call for additional control loops to restrict the maximum armature current and observe tolerances on armature voltage. Such systems fall in the catagory of multi-variable control systems. The logic governing the design of such multi-loop system is following:

The main or outer-most feed back loop is to regulate primary variable of interest while the inner control loops are used to override the error signals of enveloping loop and provide a reference signal to the immediately enclosed loop. Thus reference signal for any inner loop is obtained from the error in the

enveloping loop. In the system under consideration, below base speed of motor, the error in speed serves as reference to the generator field current. Above the base speed, the reference for motor field current is modified by over-voltage at armature terminals. The tendency of over-voltage is seen during field weakening due to steep rise in speed of motor which is due to release of stored magnetic energy.

A.3 Protective Features and Stabilising Components

The following machine protection is available

1) Protective cut-out in armature circuit provided at 2.3 p.u.

2) Over-voltage relay set at 1.2 p.u.

The stabilising components are provided to achieve safe operating limits on,

1) the maximum rate of change of field current so that punch through does not occur in the thyristors feeding the field system,

2) the maximum rate of change of armature currents to prevent arcing across brush contacts,

3) the maximum acceleration of motor to prevent breakdown in mechanical transmission.

By limiting the outputs of both the field error amplifier and the armature error amplifier to preset values, maximum rates of changes on field current and armature current can be achieved. The four saturating amplifiers are shown bearing block nos. b1, b2, b3 and b4 in Fig.A-2. Note that to obtain fast response large amplification is associated with these blocks.

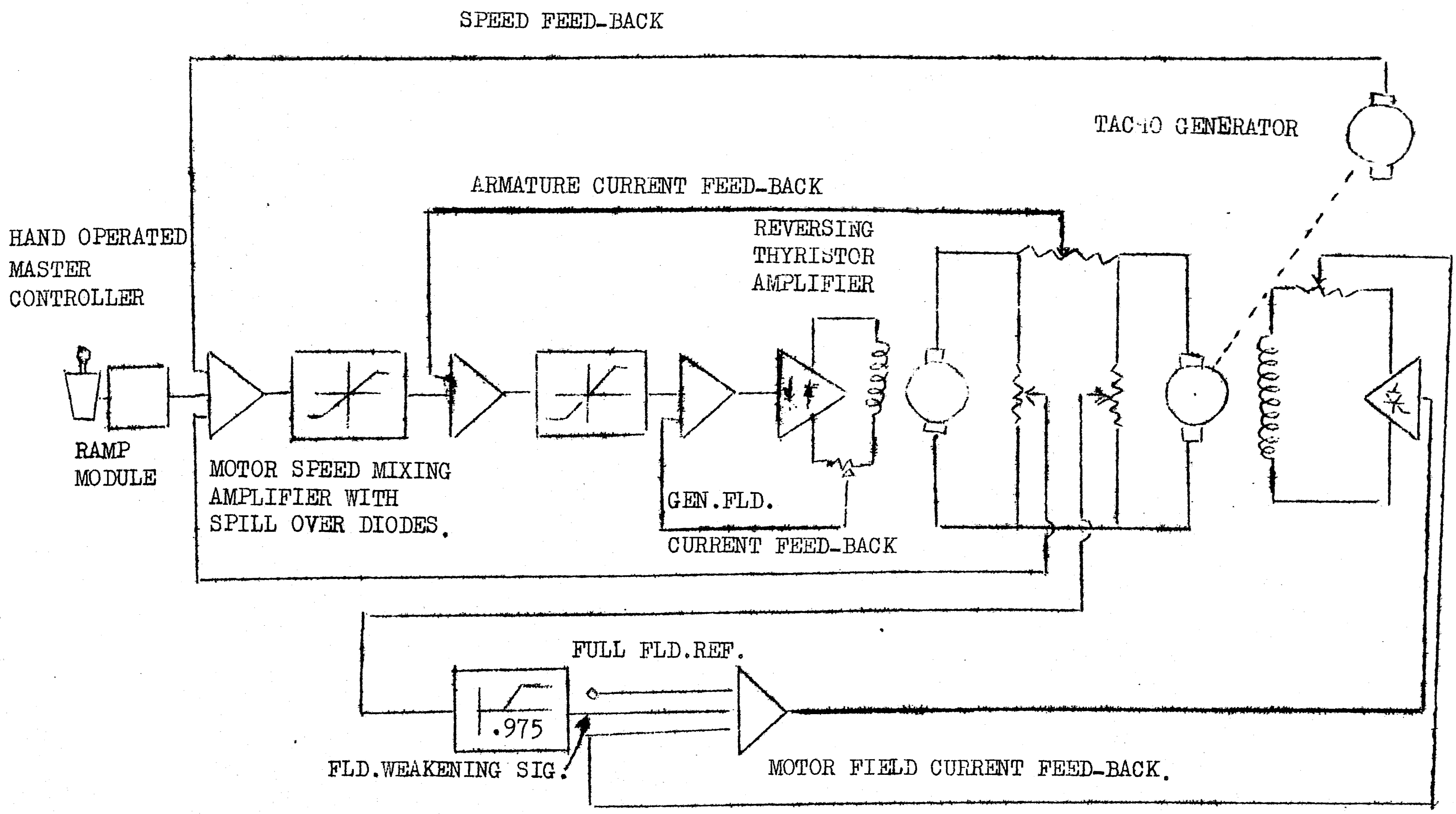


Figure A-1

The schematic with basic control loops for speed regulation.

A.4 Mathematical Model of the System

The per unit values for normalisation are same as chosen in Chapter II. With the assumptions as stated below one arrives at the block diagram as given in Fig.A-2.

Assumptions

1) The generator field characteristics is linear. This assumption does not introduce any appreciable error.

2) The generator speed is constant. Generally the ward-leonard generator is driven by a synchronous motor with over excited field, so this assumption is close to reality.

3) The time constants of triggering components are zero. The time delays introduced by triggering components are much smaller than the effective field time constants (effective time constant with forcing field is smaller than when the field is normally excited). This assumption simplifies the model.

A.5 Analog Simulation

The armature circuit and mechanical circuit are simulated by equations A.5.1

$$\left.\begin{aligned} (d\, i_a/dt) &= r_a * i_a / L_a + (e_g - e_b) / L_a \\ (d\, N_s/dt) &= K_t * i_a * \phi(i_{fm}) \end{aligned}\right\} \quad \ldots \quad \ldots \quad \ldots \quad \text{A.5.1}$$

The simulation diagram is given in Fig.A.3a,b.

The error amplifiers are simulated by a first order approximation. For an amplifier with gain of 50 and response time of 80 milli-seconds the approximated transfer function shall be $50 / (1 + p0.02)$. The corresponding simulation is shown in Fig.A-3c.

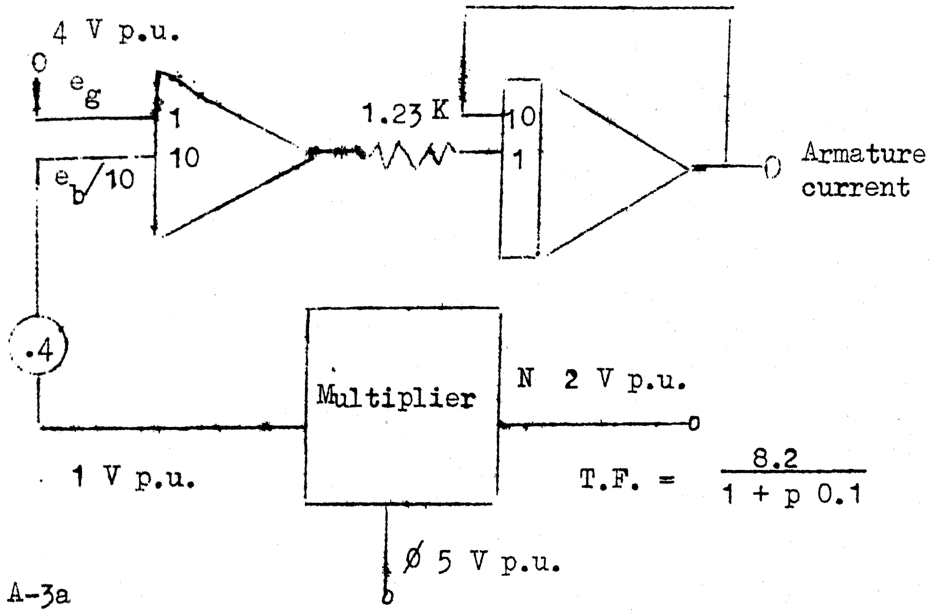
The simulation of Ramp is shown in Fig.A-3d. Here we have a saturating block for which the output is a input polarity actuated step. The step size can be changed by moving the arms of potentiometers biasing the diodes. The output of saturating block shall be of constant amplitude until the instant when integrator output attains reference amplitude. As a result, the output of integrator will be a Ramp before a matching amplitude is reached and a constant thereafter.

The motor field flux saturation is simulated by following a procedure suggested in White and Woodson[31]. The inverse saturation factor was calculated and a piecewise linear approximation required slopes of 1, 1.315 and 1.657 between points 0-0.76 p.u., 0.76-0.93 p.u. and 0.93-1 on ordinates. The analog computer simulation is shown in Fig.A-3e,f.

The complete simulation is shown in Fig.A-3g.

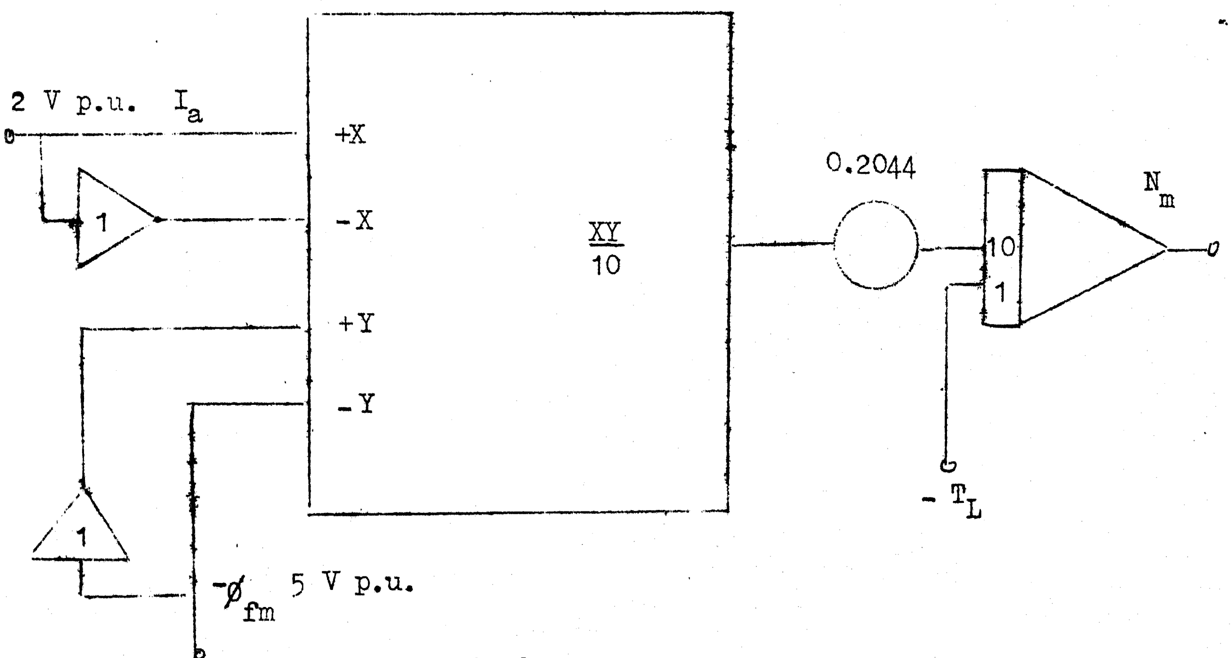
Simulation Results

As mentioned earlier there are four basic control loops, one each for the generator field current, armature current, speed and motor field current. For the first three the error in enveloping loop provides the reference. The compensating parameters can be computed



A-3a

A-3b



Note: Analog Computer used : TR 20,

For gain = 1 $R_i = R_f = 100$ K , For gain = 10 $R_i = 10K$, $R_f = 100K$.

Figures A-3a,b

Simulation of armature circuit (A-3a), Simulation of Mechanical Circuit (A-3b).

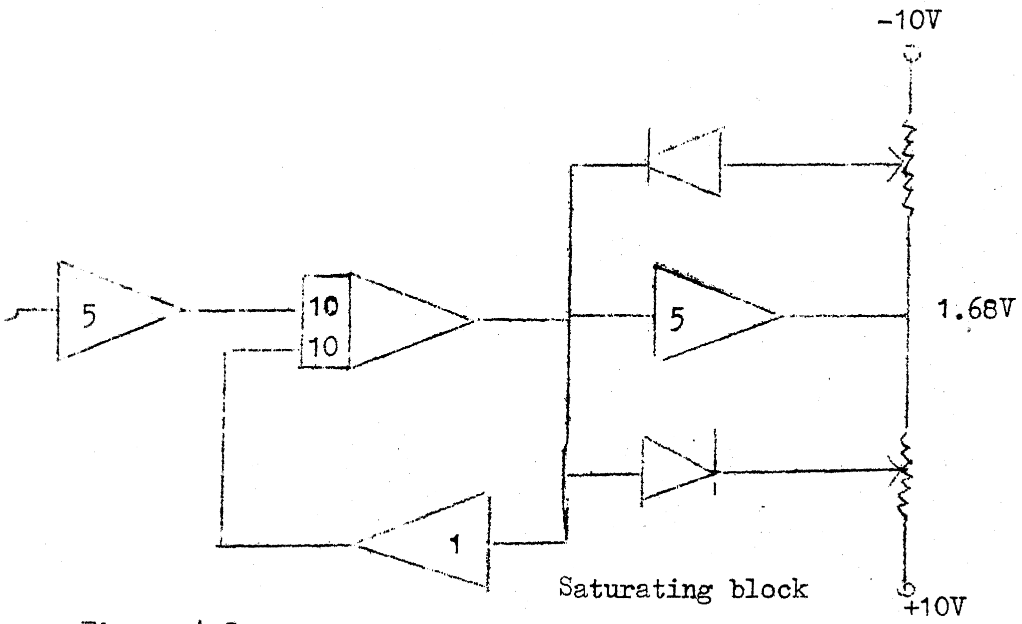


Figure A-3c
Simulation of error amplifiers.

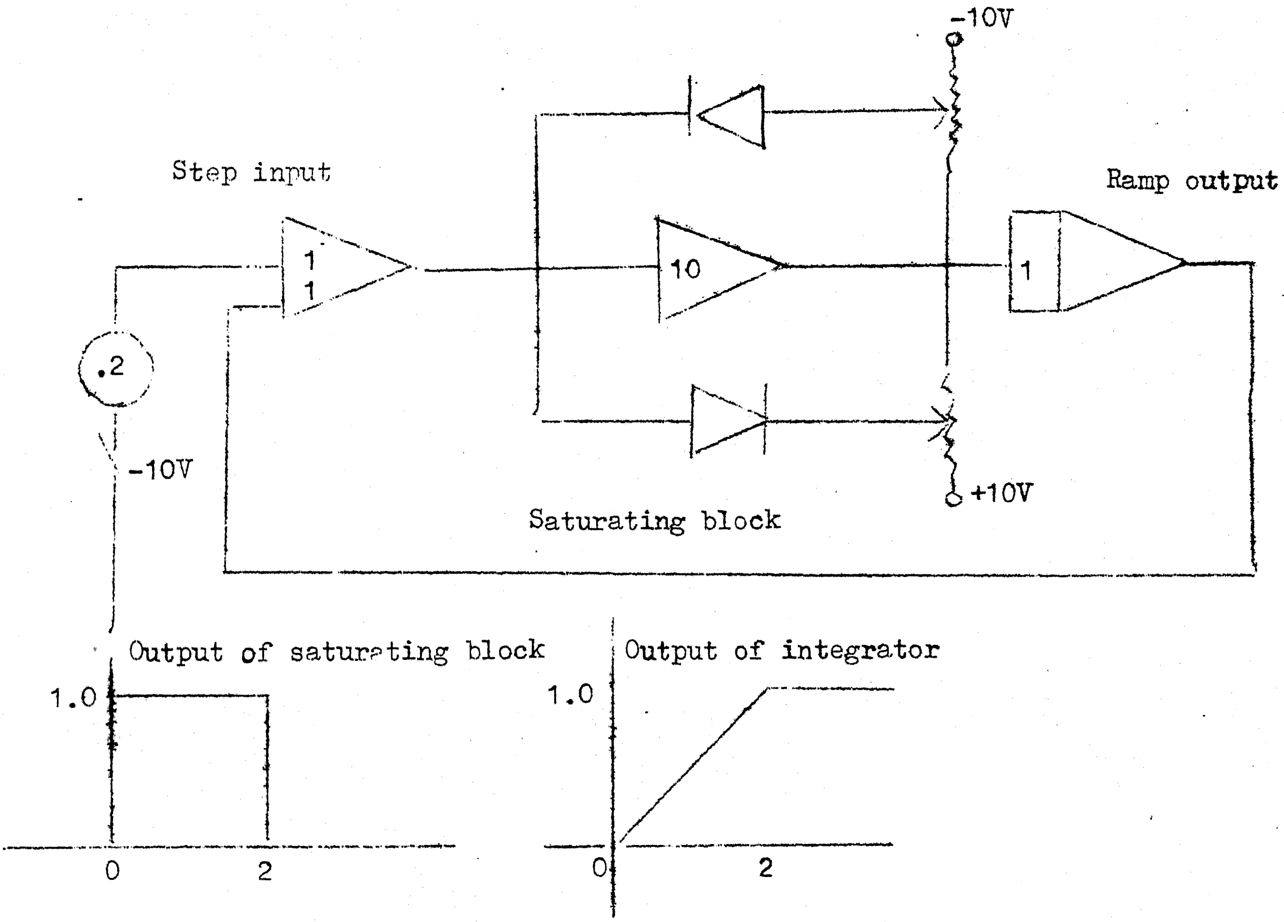


Figure A-3d

Simulation of Ramp.

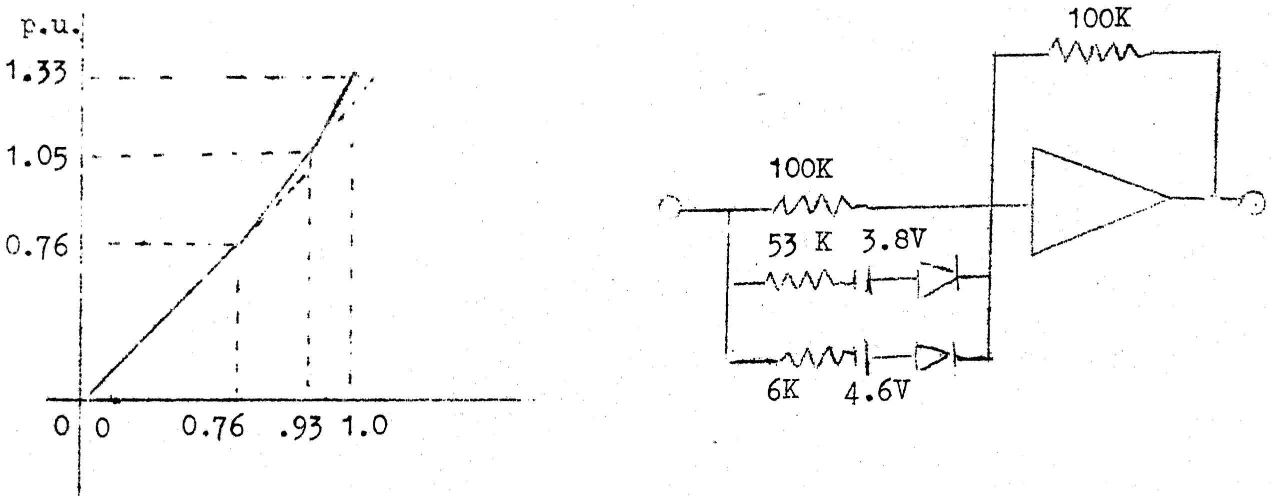


Figure A-3e

Simulation of inverse saturation curve.

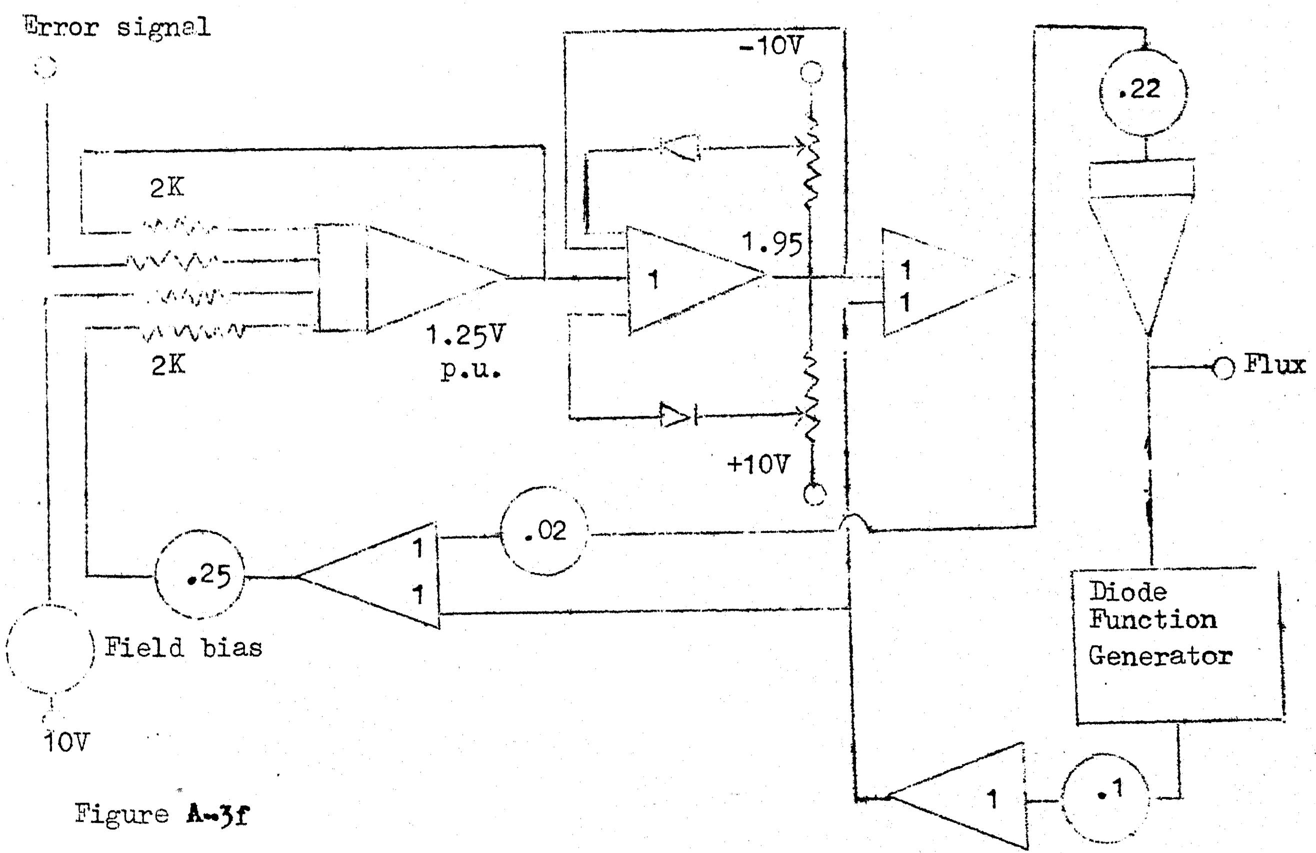


Figure A-3f

Simulation of motor field control circuit.

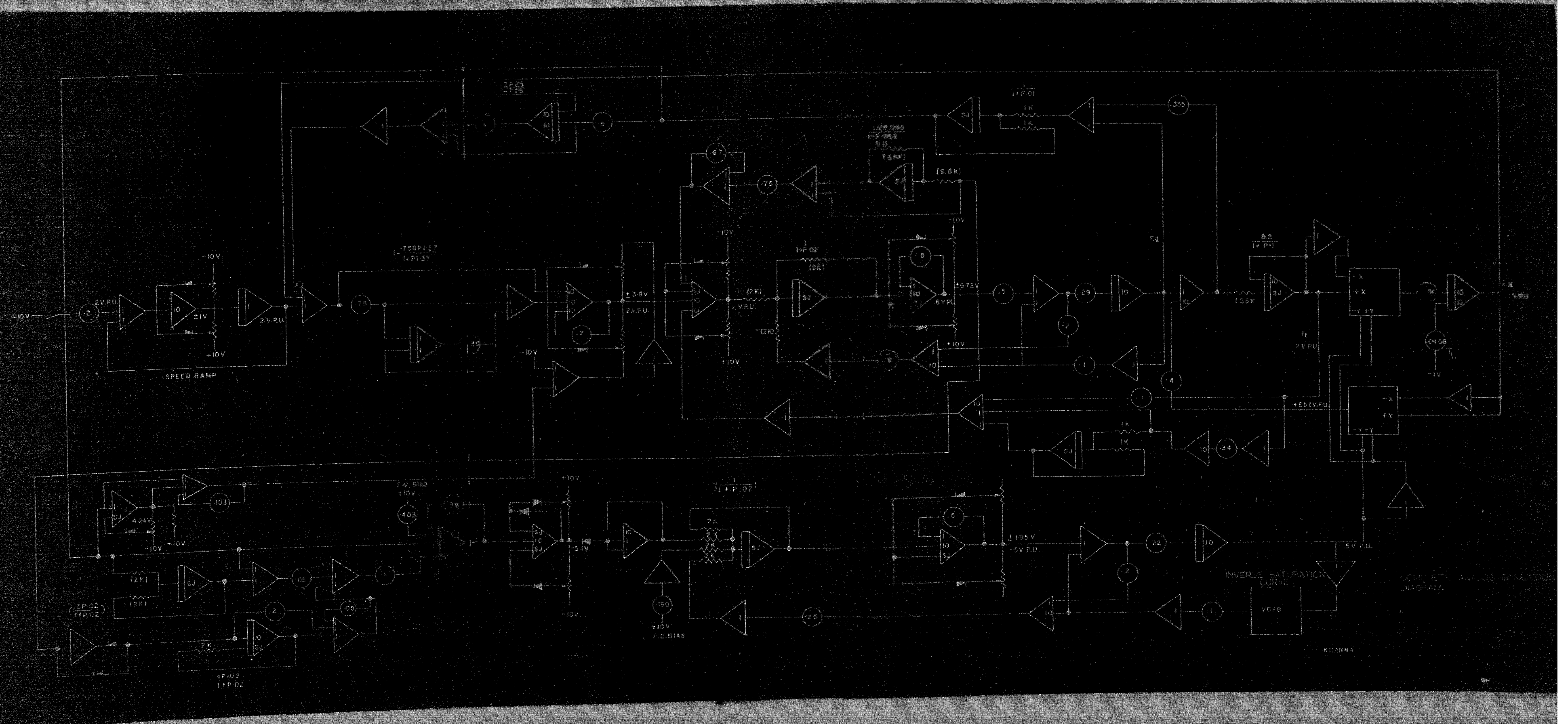


Figure A-3g

by following a sequential testing procedure as described below. In each step some useful results is obtained. In the first three tests described below the motor field flux is held at 1 p.u. to simulate generator field control mode.

Step-1

We open all but the inner most loop (in our case Gen.Fld.Curr. loop) and apply a step input as reference. This value should preferably be larger than that of a unit step so that one is assured that voltage amplitudes are well below saturation level of amplifiers during transients in the total simulation. Fig.A-4 depicts the field current response for a step size of 1.75 p.u. step input. In block diagram (Fig.A-2) this input corresponds to input to field mixing amplifier (Block A) and physically corresponds to a step error in armature current. This response is satisfactory and does not require any compensation.

Step-2

We now test the enveloping loop (armature current loop) by applying a step input as reference and the procedure of step 1, is repeated. The uncompensated response is shown in Fig.A.5. We find that the armature current response has large overshoot. To improve the response a feedback signal through a phase lead compensating network (known as phase advancer in industry) is given. The parameters of compensating block are adjusted to obtain satisfactory response for both large and small step reference inputs.

Step-3

Next the performance of speed control loop is checked. It was found necessary to increase voltage feedback gain to 200 to ensure little overshoot and regenerative current. Next this circuit is tested for Ramp input (Fig.A-6a,b).

Step-4

To check for field weakening the circuit is connected completely and reference signal corresponding to run to top speed is given. The compensating parameters in the feedback loop from the terminal voltage are obtained for no load. It was felt that an additional loop with phase advance will have to be included to reduce speed drop on impact loading and load throwoff (ref. Fig.A-7a-d).

Step-5

A test run with 1.5 p.u. full load is taken to ensure that during short term and accidental overloads the mill does not fail to operate.

The summary of the overall performance is given in Table A-1.

Step 1

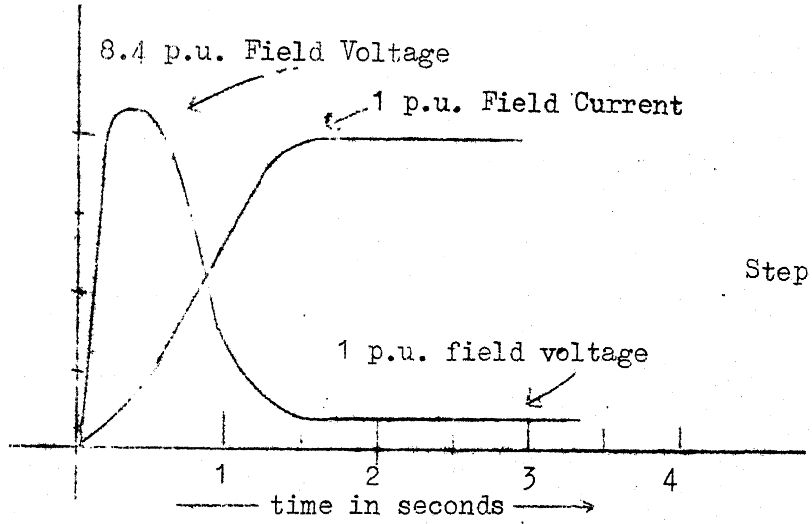


Figure A-4
Response of the generator field circuit.

Step 2

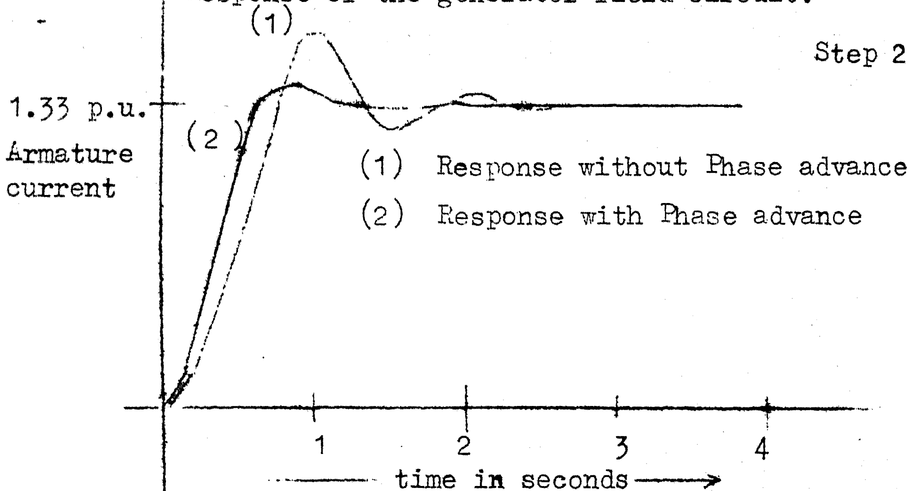


Figure A-5
Current circuit response.

Step 3

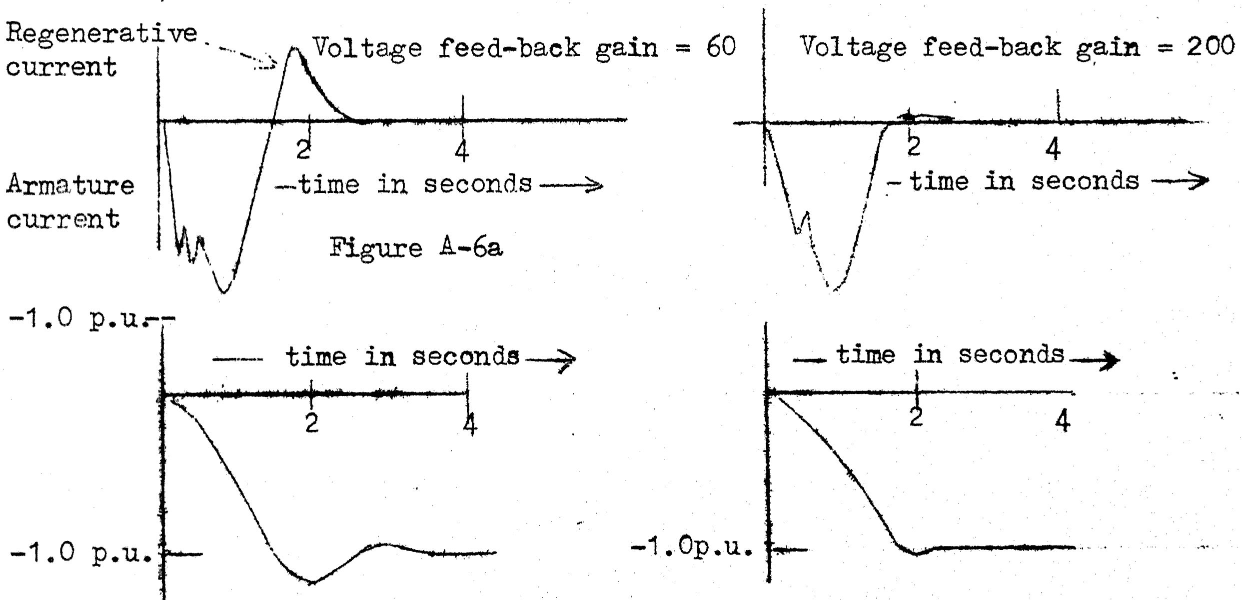


Figure A-6
Speed and current response to a step input.

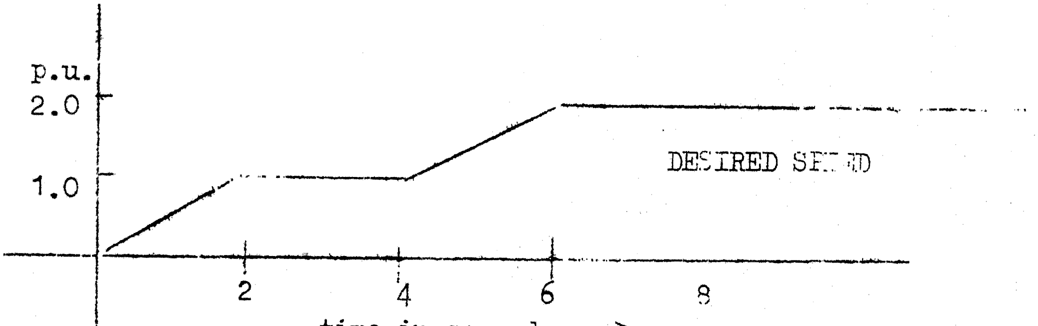


Figure A-7a

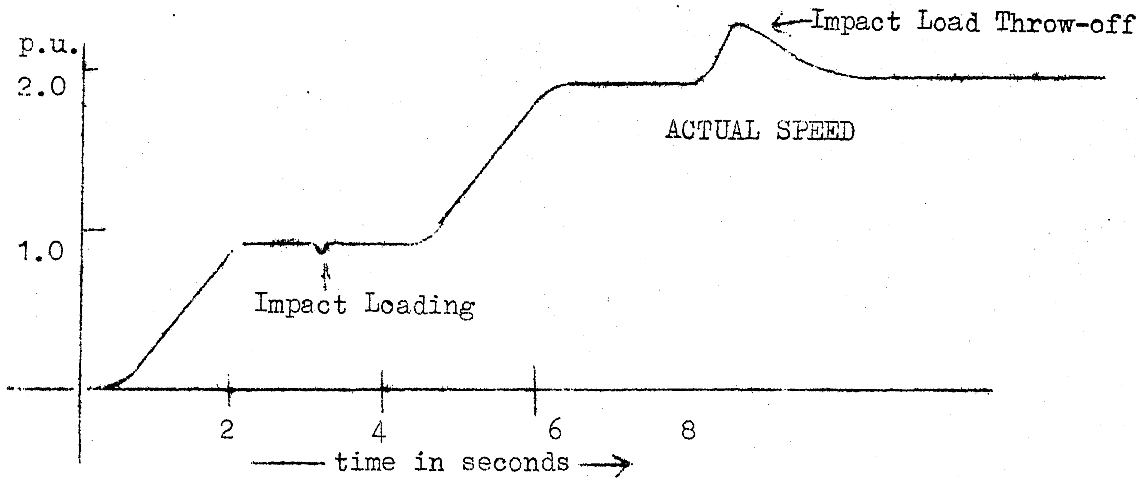


Figure A-7b

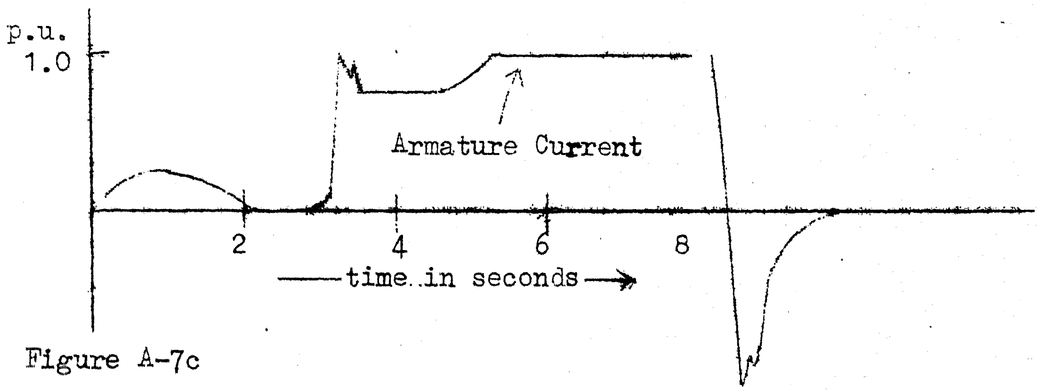


Figure A-7c

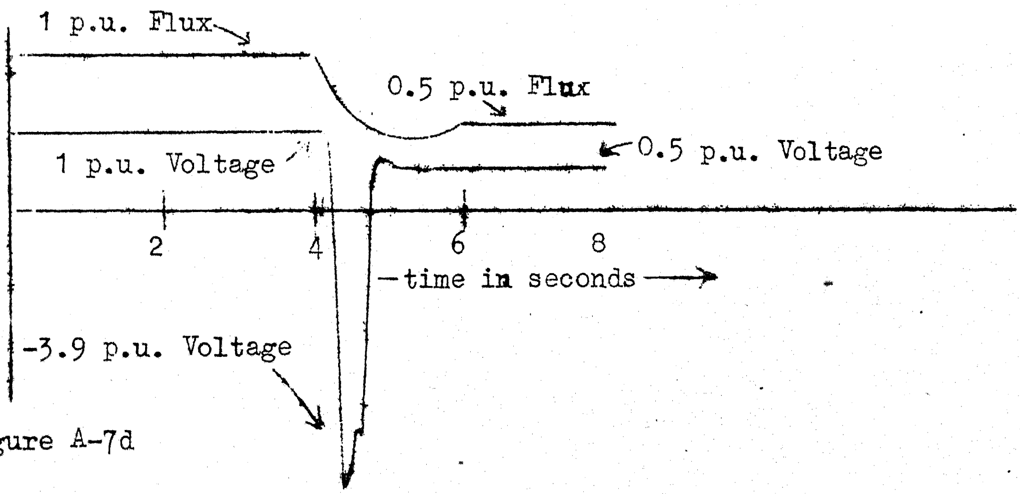


Figure A-7d

Table A-1

Gen.Fld.Ckt. Time Const.	With forcing function	Without forcing function
	0.1875 seconds	3.44 seconds

Armature Circuit Overshoot (Step Input)		With phase advance	Without phase advance
	Large step	5.6%	17.6%
	Small step	1.5%	8%

Speed loop performance (Step Input)			
	Term.Vol.Feedback gain	200	60
	Overshoot	almost 0%	25%
	Regenerative current	0.02 p.u.	0.4 p.u.

With weak field	Impact	Recovery time (Less than)	Arm. Current (p.u.)	Arm.Curr. peak (p.u.)	Term. Voltage %	S.S.Top speed (p.u.)
Load On	10%	1.5 sec.	0.98	1.1	10	1.9
Load Throwoff	30%	3 sec.	avg.0	1.8	10	2.1

APPENDIX - B

The Nature of Stalling Torque during Rolling

During rolling, the mill drive is required to overcome torque due to the metal slippage past the arc of contact (Fig.B-1a) with rolls and deformation of steel which is in plastic condition. The mill drive experiences this torque as a stalling torque[8,11] which is evident from the theoritical torque vs. time characterstics shown in Fig.B-1b. The gentle upward slope in the characteristics of Fig.B-1b indicates higher torque requirement as rolling proceeds primarily because of cooling (radiation loss) of the metal. It is assumed throughout this investigation that torque vs. time characteristics are prespecified by a process computation from slab dimension and furnace data. However, for the purpose of present simulation the characteristics as shown in Fig.B-1c has been used.

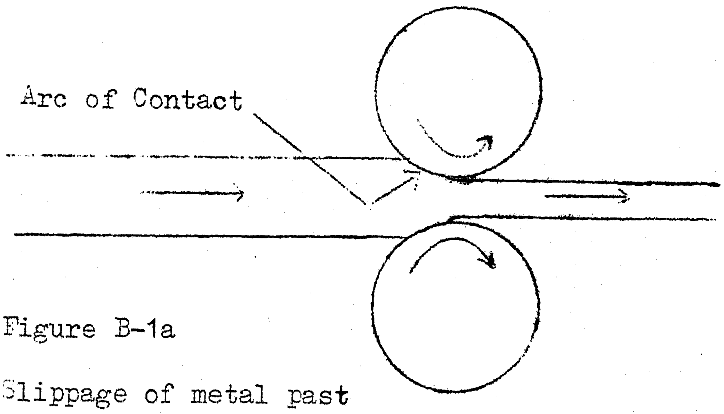


Figure B-1a

Slippage of metal past
the arc of contact.

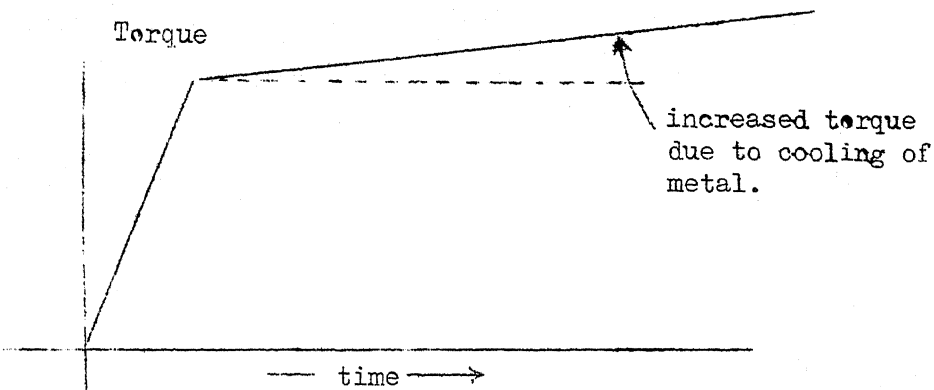


Figure B-1b

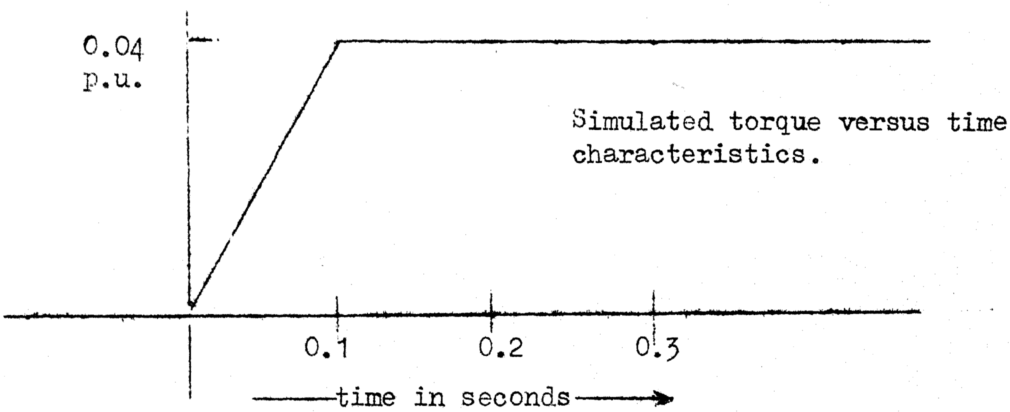


Figure B-1c

Figure B-1

The nature of torque during rolling.

APPENDIX - C

The Derivation of Bellman's Functional Equation

Let the state equations for the system be

$$\dot{x} = f(x, u, t) \qquad \text{C.1.1}$$

where x is $n \times 1$ vector defining the state of the system

u is $r \times 1$ vector defining the forcing functions

$n \leq r$

$x(0) = x(0)$, $t \in (0, T)$

It is desired to extremise

$$E = \int_0^T F(x, u, t)\, dt \qquad \text{C.1.2}$$

For such problems, Bellman has suggested a direct computational procedure which is a consequence of imbedding principle and optimality principle.

Imbedding Principle[2]: A problem with a fixed initial state and a fixed interval of operation may always be viewed as a special case of a more general problem, with a variable initial state and variable interval of operation.

If t' is such that $0 \leq t' \leq T$ then the original problem can be considered as a particular case of a more general problem where $x(t')$ defines the initial state and (t', T) defines the interval of operation.

Optimality Principle[2]: Any choice of u which is optimal over the interval $(0, T)$ is necessarily optimal over any sub-interval (t', t'')

where $0 \leqslant t' \leqslant t'' \leqslant T$.

Thus the optimum value of E is a function of initial state and time.

$$\overset{o}{E}(x(0),\ 0) = \text{optimum } E = \min_{u \in U} \left[\int_0^T F(x,\ u,\ t)\ dt \right] \qquad C.1.3$$

Then by invoking optimality principle and imbedding principle, we have

$$\overset{o}{E}(x(0),0) = \min_{u \in U} \int_0^{t'} \left[F(x,u,t)\ dt + \overset{o}{E}(x(t'),\ t') \right] \qquad C.1.4$$

Let $t' = \Delta t$, then because $\overset{o}{E}$ is a continuous function of x and t we have

$$\overset{o}{E}(x(t'),t) = \overset{o}{E}(x(0),0) + \left[(\partial \overset{o}{E}(x(0),0)/\partial t) + (\partial \overset{o}{E}/\partial x) * \dot{x} \right] \Delta t$$

$$+ \text{ higher order terms} \quad .. \quad .. \qquad C.1.5^*$$

The substitution of C.1.5 in C.1.4 yields the necessary condition for optimality C.1.6

$$\min_{u \in U} \left[(\partial \overset{o}{E}/\partial t) + (\partial \overset{o}{E}/\partial x) * \dot{x} + F \right] = 0 \quad .. \qquad C.1.6$$

The equation C.1.6 is known as Bellman's functional equation.

---

* Transposition is implied when ever necessary.